



UNIVERSIDAD
COMPLUTENSE
MADRID

Proyecto de Innovación

Convocatoria 2019/2020

Nº de proyecto

335

Título del proyecto

Creación de un paquete de R para gestionar bases de datos de preguntas de test
compatibles con Moodle

Nombre del responsable del proyecto

JOSÉ FERNÁNDEZ MENÉNDEZ

Centro

Facultad de CC. Económicas y Empresariales

Departamento

Organización de Empresas y Marketing

Economía Financiera y Actuarial y Estadística

1. Objetivos propuestos en la presentación del proyecto

El objetivo del proyecto consistía en desarrollar una serie de herramientas de software que complementen las capacidades de Moodle para la gestión de bases de datos de preguntas de test. Estas herramientas se integrarían en un paquete de R en el que se pretendía conseguir un máximo de usabilidad para el usuario final. Todas las herramientas utilizadas y desarrolladas serán software libre.

Las funcionalidades concretas que se pretendían implementar eran:

- capacidad para crear de forma sencilla bases de datos *sqlite* con preguntas de test
- capacidad para introducir de forma cómoda preguntas de test clasificadas por temas en dichas bases de datos utilizando formularios implementados con TclTk, que puede ser usado desde R o como una aplicación independiente.
- capacidad para generar informes a partir de las bases de datos de preguntas de test que permitan revisar con facilidad dichas preguntas, modificarlas y corregirlas, moverlas de un tema a otro, anularlas, etc
- capacidad para exportar las preguntas de la base de datos al formato *gift* de Moodle, para que puedan ser importadas desde Moodle con un simple click
- capacidad para generar exámenes de test en papel, en distintos formatos más o menos customizables, para facilitar el uso alternativo de exámenes de test en papel y en el campus virtual según las necesidades y circunstancias de cada momento

Con esto lo que se pretendía es que toda la gestión de las preguntas de test se llevase a cabo de forma cómoda fuera del campus virtual, es decir, de forma *off-line*, manteniendo las preguntas en un archivo de base de datos *sqlite* que pueda ser accedido fácilmente desde distintas aplicaciones. De esta manera, cuando se deseara realizar un test en el campus virtual bastaría con importar en Moodle las preguntas de test exportadas desde la base de datos.

Todo el software desarrollado debería estar disponible en repositorios públicos.

2. Objetivos alcanzados

Los objetivos propuestos se han conseguido parcialmente, pero de forma razonablemente satisfactoria.

Se ha programado una aplicación en TclTk que permite de forma cómoda crear y gestionar preguntas de test que pueden ser exportadas fácilmente al formato *gift* de Moodle, con lo cual pueden ser importadas a la base de preguntas del campus virtual de forma inmediata.

También se ha escrito una función en R que permite, a partir de una base de datos de preguntas, generar exámenes de test en papel de acuerdo con una serie de formatos predefinidos en latex (de momento tres formatos, para exámenes con 10, 15 y 20 preguntas, pero que pueden ampliarse y modificarse con relativa facilidad).

La aplicación en TclTk es plenamente funcional y puede ser ejecutada tanto en Linux como en Windows y en Mac si se tiene instalado un intérprete de TclTK, por ejemplo el que proporciona de forma gratuita la empresa ActiveState (<https://www.activestate.com/products/tcl/downloads/>).

Quedaría añadir a la aplicación en TclTk la capacidad de exportar los enunciados de las preguntas a pdf, para poder imprimirlas y revisarlas con comodidad, añadir alguna funcionalidad adicional, integrar toda esta funcionalidad en un paquete de R, y crear una cuenta en github desde la que gestionar el desarrollo de la aplicación y su distribución.

3. Metodología empleada en el proyecto

La metodología seguida en el proyecto ha consistido en primer lugar en revisar una serie de programas de ordenador basados en software libre que guardan relación o proporcionan una funcionalidad similar en algun medida con lo que aquí se propone. En concreto el paquete moodle para latex y los paquetes de R *exams*, *RndTexExams* y *TexExamRandomizer*.

También se ha revisado una gran cantidad de documentación sobre la base de datos sqlite, sobre R y sobre el lenguaje de programación TclTk, que son las herramientas fundamentales que se han usado en el proyecto.

Tras esto se han ido elaborando una serie de módulos de software que posteriormente se han integrado (de forma parcial) y se ha elaborado un pequeño tutorial que describe el funcionamiento de la aplicación desarrollada.

4. Recursos humanos

Han participado en la realización de este proyecto los siguientes profesores:

Del Departamento de Organización de Empresas y Marketing:

MARTA FOSSAS OLALLA

BEATRIZ MINGUELA RATA

ANTONIO MATEO RODRIGUEZ DUARTE

FRANCESCO DOMENICO SANDULLI

Del Departamento de Economía Financiera y Actuarial y Estadística:

MARIA ZULEICA DIAZ MARTINEZ,

además del responsable del proyecto, JOSÉ FERNÁNDEZ MENÉNDEZ, del Departamento de Organización de Empresas y Marketing.

Todos ellos son profesores permanentes de la UCM, con una amplia experiencia docente en sus respectivas áreas de conocimiento, y que imparten y coordinan un gran número de asignaturas, tanto de grado como de máster. En la impartición de estas asignaturas utilizan de forma habitual y desde hace años el campus virtual y la herramienta de cuestionarios online que ofrece la plataforma Moodle, con lo cual están

sobradamente familiarizados con las ventajas y las carencias e inconvenientes que su uso plantea. También están familiarizados con, y utilizan habitualmente en sus tareas docentes e investigadoras, diversas tecnologías de la información y las comunicaciones y herramientas de software y de programación, R entre otras.

5. Desarrollo de las actividades

En primer lugar se ha llevado a cabo una revisión exhaustiva de documentación técnica necesaria sobre lenguajes de programación necesarios para relanzar el proyecto.

Especialmente se han consultado, además de abundante documentación on-line, los siguientes manuales:

The Art of R Programming (2011), Norman Matloff. No starch press, San Francisco.
Hands-On Programming with R (2014), Garrett Grolemund. O'Reilly, Sebastopol, CA.
Tcl/Tk. A Developer's Guide, 3ª ed (2012), Clif Flynt. Morgan Kaufmann, Waltham, MA.
The Tcl Programming Language. A comprehensive Guide (2017), Ashok P. Nadkarni. Amazon Distribution.

Using SQLite (2010), Jay A. Kreibich. O'Reilly, Sebastopol, CA.

The Definitive Guide to SQLite (2010), Grant Allen y Mike Owens. Apress, NY.

El libro de LATEX (2003), Bernardo Cascales Salinas y otros. Pearson Prentice Hall, Madrid.

Latex and Friends (2012), M. R. C. van Dongen. Springer, Berlin.

Introducing Markdown and Pandoc (2019), Thomas Mailund. Apress, NY.

The TikZ and PGF Packages (2020), Till Tantau. Disponible en

<http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf>

Posteriormente se ha diseñado la estructura de la base de datos sqlite usada para contener las preguntas de test y se ha implementado un script en TclTk para generar bases de datos con esa estructura.

Tras esto se han desarrollado otros scripts en TclTk para gestionar las bases de datos generadas, en concreto para introducir en ella preguntas, de una en una o importadas en bloque desde un archivo de texto plano, para mostrar las preguntas por temas, para crear nuevos temas y renombrarlos en la base de datos y para generar los archivos gift.

Posteriormente se han integrado los diversos módulos en una única aplicación (denominada “la ñapa”) y se han escrito una función en R y unas plantillas en latex para generar exámenes de test en papel a partir de las preguntas contenidas en las bases de datos.

6. Anexo 1: Código fuente

Se adjunta a continuación el código fuente de la ñapa y de los archivos de R `generarTest.R`, `plantillaTest.Rnw`, `plantillaFinal15.Rnw` y `plantillaFinal20.Rnw`

```

#!/usr/bin/tclsh

encoding system utf-8

# esto es un programa de tcl
# será la aplicación gorda que lo integra todo
# tiene 4 frames
# la de arriba indica la base de datos
# la siguiente es para seleccionar la base de datos
# las otras tiene los botones con las opciones

# v.02: meto un toplevel para lo de gestionar temas
# v.03: intento escribir en la base de datos usando una transacción y meto
código para generar gifts
# v.04: corrijo erratas en gestionar temas y meto lo de mostrar temas en un
widget text
# v.05: meto el procedimiento para mostrar preguntas con el widget text
# v.06: meto el procedimiento para editar preguntas
# v.07: cambio la escritura de los gifts para escapar los caracteres especiales
de moodle
#      esos caracteres son ~ = # { } :
#      la forma de escaparlos es con regsub -all {(:#|{|}|~|=)} $variable {\%
1}
# v.08: intento que lo de guardar en la base de datos (tras editar las
preguntas) funcione bien
# v.09: pongo que las erróneas resten 1/3;
#      lo de las correctas no lo cambio pq se supone que sólo habrá 1
# v.10: esta usa markdown en lugar de (o además de) moodle o html
#      -de momento cambio el procedimiento de guardar las nuevas preguntas
#      -luego el procedimiento de mostrar las preguntas por temas
#      habrá que chequear si existe la columna markdown en la base de datos
#      y si cada pregunta es o no con markdown
#      si sí lo es el único cambio será que en el enunciado aparece al pcpio
[mk]
#      -cambio también el procedimiento para generar gifts
# v.11: habilito el botón de generar base de datos (genera una vacía)
# v.12: añadido un botón para importar preguntas desde .txt
# v.13: hago que al generar una base de datos, ésta quede seleccionada, y que me
lance un aviso de que todo ok!
# v.14: igual que la v.13 pero con la puntuación de Compras
# v.15: meto la instrucción de encoding al principio
#      retoco lo de incluir una quinta pregunta
# v.16: corrijo errores en el procedimiento para editar preguntas a mano
# v.17: pongo radiobuttons para seleccionar cuánto restan las erróneas

package require Tk
package require sqlite3
package require fileutil

set cfond "pale green"
set cbad "sandy brown"
set ii 0; # esto es una prueba
set listaii {}
set cambiosOK true
set nuevoTEMA false
set preg {}
set maxfilas {}
set numtemas {}

frame .fsup
frame .falt -bg firebrick1 -bd 2
frame .fmed
frame .fmedbajo

```

```

frame .finf

# tipos de ficheros para la selección

set tipos {
    {{Bases de Datos} {.db}}
    {{Todos} {*}}
}

set tiposTxt {
    {{Archivos de texto} {.txt}}
    {{Todos} {*}}
}

# base será la variable global que contiene el nombre de la base de datos
# seleccionada en cada momento
# ruta_base contendrá la ruta completa de la base seleccionada
set base "no hay base seleccionada"

# base_ok indica si hay seleccionada una base de datos adecuada
# habrá que activar/desactivar botones según su valor
set base_ok false

# en la frame superior pongo dos labels que indican la base de datos
# seleccionada
label .fsup.letiqueta -text "Base de Datos: " -height 3 -width 30
label .fsup.lbase -textvariable base -height 3 -width 30 -background white
pack .fsup.letiqueta .fsup.lbase -side left

# en la frame alta pongo un botón para seleccionar la base de datos
# y otro para crearla
button .falt.bselecc -text "Seleccionar base de datos" -width 30 -height 3 -
command {
    set candidata [tk_getOpenFile -initialdir [pwd] -filetypes $tipos]
    if {$candidata ne ""} {
        if {[regexp -nocase sqlite3 [fileutil::fileType $candidata]]} {
            set base_ok true
            set ruta_base $candidata
            set base [file tail $candidata]
            if {[regexp dpdg [info commands dpdg]]} {dpdg close}
            sqlite3 dpdg $ruta_base -create false -readonly false
            set mkyes [dpdg eval {SELECT COUNT(*) FROM
pragma_table_info('preguntas') WHERE name='markdown'}}
        }
    }
}

button .falt.bcrear -text "Crear nueva base de datos" -width 30 -height 3 -
command {crear}
pack .falt.bcrear .falt.bselecc -side left

# en la frame del medio alto pongo los botones de meter datos y ver datos
button .fmed.bmeter -text "Introducir Preguntas a mano" -width 30 -height 3 -
command {editar}
button .fmed.bmostrar -text "Mostrar Preguntas" -width 30 -height 3 -command
{mostrar}
pack .fmed.bmeter .fmed.bmostrar -side left

# en la frame del medio bajo pongo un botón de Introducir preguntas desde .txt
button .fmedbajo.bmettxt -text "Introducir Preguntas desde .txt" -width 30 -
height 3 -command {meterTxt}
pack .fmedbajo.bmettxt

```

```

# en la frame inferior meto los botones de generar gift y gestionar temas
button .finf.bgift -text "Generar GIFT" -width 30 -height 3 -command {generar}
button .finf.btemas -text "Gestionar Temas" -width 30 -height 3 -command {temas}
pack .finf.bgift .finf.btemas -side left

# al arrancar buscamos una base de datos válida en el directorio actual
set lista_bases [glob -nocomplain *.db]

# de momento no chequeo la base de datos con una pragma, pero podría hacerlo
# sólo compruebo con fileType que es sqlite3
set tiempo 0
foreach candidata $lista_bases {
    set tt [file atime $candidata]
    if {$tt > $tiempo} {
        if {[regexp -nocase sqlite3 [fileutil::fileType $candidata]]} {
            set base_candidata $candidata
            set tiempo $tt
            set base_ok true
            # puts "[set base_ok]"
        }
    }
}

if {$base_ok} {
    set base $base_candidata
    set ruta_base [file normalize $base_candidata]
    sqlite3 dpgd "$ruta_base" -create false -readonly false
    # puts "$ruta_base"
    # puts "existe dpgd: [info exists dpgd]"
    set mkyes [dpgd eval {SELECT COUNT(*) FROM pragma_table_info('preguntas')
WHERE name='markdown'}}]
}

# tendré que comprobar que hay una base de datos seleccionada para habilitar
botones

#####
#### procedimiento para meter preguntas desde fichero .txt #####
#####

proc meterTxt {} {

    global tiposTxt file_txt ruta_txt mkyes

    if {![regexp dpgd [info commands dpgd]]} then {
        tk_messageBox -message "no hay una base de datos seleccionada" -type ok
        return 1
    }

    toplevel .meterTxt
    grab set .meterTxt
    wm attributes .meterTxt -topmost true

    set file_txt "no hay archivo .txt seleccionado"

    # buscamos archivos .txt en el directorio actual
    # compruebo que son tipo text y selecciono en principio el más reciente
    set txt_ok false
    set lista_txt [glob -nocomplain *.txt]
    # puts $lista_txt
    set tiempo 0
    foreach candidato $lista_txt {
        set tt [file atime $candidato]
        if {$tt > $tiempo} {

```



```

        if {[regexp -nocase text [fileutil::fileType $candidato]]} {
            set txt_candidato $candidato
            set tiempo $tt
            set txt_ok true
            #puts "[set txt_ok]"
        }
    }
}

if {$txt_ok} {
    set ruta_txt [file normalize $txt_candidato]
    set file_txt [file tail $ruta_txt]
}

# puts "$file_txt"

# creo dos frames
# una superior para seleccionar el archivo .txt
# una inferior con el botón de importar
frame .meterTxt.ftop
frame .meterTxt.flow

button .meterTxt.ftop.bseleccionar -text "Seleccionar archivo .txt" -width
35 -height 2 -command {
    set fileTxt [tk_getOpenFile -initialdir [pwd] -filetypes $tiposTxt]
    if {$fileTxt ne ""} {
        if {[regexp -nocase text [fileutil::fileType $fileTxt]]} {
            set txt_ok true
            set ruta_txt [file normalize $fileTxt]
            set file_txt [file tail $ruta_txt]
        }
    }
}

# al final tendré en $ruta_txt la dirección del fichero con las preguntas en
texto plano

label .meterTxt.ftop.lseleccionar -textvariable file_txt -width 35 -height 2
-background white

pack .meterTxt.ftop.bseleccionar .meterTxt.ftop.lseleccionar -side left

button .meterTxt.flow.bimportar -text "Importar .txt" -width 40 -height 3 -
command {
    set ficheroTexto [open $ruta_txt r]

    set marcadorTema false
    set contadorLineas 0
    set contador 0
    set dicPreguntas [dict create]
    # en este diccionario cada key es un nº de tema
    # y los valores asociados son listas con las preguntas del tema
    set mdw 0
    set anulada 0
    set obsoleta 0

    while {[set nc [gets $ficheroTexto linea]] != -1} {
        if {![string is space $linea]} then {
            if {[regexp -nocase {^[:space:]*tema[:space:]}+([[:digit:]]+)[[:space:]]*$} $linea tem ntem]} {
                # puts ok
                if {$contadorLineas > 0 && $contadorLineas < 5} {dict lappend
dicPreguntas $ntem $listona}
                set marcadorTema true
            }
        }
    }
}

```

```

        set contador 0; # cuando entro en un tema
        set contadorLineas 0; # para contar líneas de 5 en 5
        set ntem [expr $ntem]; # uso expr para que un 01 se quede como 1
    } else {
        # si no hay un nº de tema al pcpio:
        if {! $marcadorTema } {
            if {[tk_messageBox \
                -message "No se indica tema al principio" \
                -type ok \
                -parent .meterTxt] eq "ok"} { return }
        }
        incr contador
        set contadorLineas [expr {$contador % 5}]
        # listona será una lista con todas las preguntas y respuestas
asociadas al tema
        lappend listona $linea
        # cuando contadorLineas llega a 0 añadimos la lista al
diccionario
        # (será una pregunta y sus 4 respuestas)
        if {$contadorLineas == 0} {dict lappend dicPreguntas $ntem
$listona; set listona {}}
    }
}

close $ficheroTexto; # ya he leído todo lo que necesito
# aquí hay que comprobar que contadorLineas está a 0, si no abortamos
# mejor simplemente escribo en el diccionario lo que haya y luego ya
chequeo
if {$contadorLineas > 0 && $contadorLineas < 5} {dict lappend dicPreguntas
$ntem $listona}
# puts $dicPreguntas

set numeroPreguntas 0
set lkeys [dict keys $dicPreguntas]

foreach keytema $lkeys {
    foreach pregetema [dict get $dicPreguntas $keytema] {
        incr numeroPreguntas
        if {[llength $pregetema] != 5} {
            if {[tk_messageBox \
                -message "Archivo de preguntas con estructura no
válida" \
                -type ok \
                -parent .meterTxt] eq "ok"} { return }
        }
    }
}

# si se llega aquí es que el fichero de texto plano con preguntas está
razonablemente bien
# en ese caso lo guardamos en la base de datos
# habrá que comprobar si los números de los temas ya están en la tabla
temas

set numeroPreguntasCheck 0

foreach keytema $lkeys {
    if {![dpdg exists "SELECT 1 FROM temas WHERE numero=$keytema"]} {
        set tit "Tema_$keytema"
        # puts $tit
        dpdg eval " INSERT INTO temas (numero,tituloTema) VALUES

```

```

($keytema,'$tit') "
}
set numTemaID [dpdg eval "SELECT id FROM temas WHERE numero=$keytema"]

foreach pregtema [dict get $dicPreguntas $keytema] {
    set enunciado [lindex $pregtema 0]
    set resp1 [lindex $pregtema 1]
    set resp2 [lindex $pregtema 2]
    set resp3 [lindex $pregtema 3]
    set resp4 [lindex $pregtema 4]

    set enunciado [regsub -all {(^([[:space:]]*\***)*)|((\
**[[:space:]]*)*)$)} $enunciado {}];# esto no debería hacer falta
    if { [ regexp {(^([[:space:]]*\***)*)|((\**+[:space:]]*)+$)} $resp1 ]
} then {
    set resp1 [regsub -all {(^([[:space:]]*\***)*)|((\
**[[:space:]]*)*)$)} $resp1 {}]
    set ok1 1
} else {
    set resp1 [string trim $resp1]
    set ok1 0
}

    if { [ regexp {(^([[:space:]]*\***)*)|((\**+[:space:]]*)+$)} $resp2 ]
} then {
    set resp2 [regsub -all {(^([[:space:]]*\***)*)|((\
**[[:space:]]*)*)$)} $resp2 {}]
    set ok2 1
} else {
    set resp2 [string trim $resp2]
    set ok2 0
}

    if { [ regexp {(^([[:space:]]*\***)*)|((\**+[:space:]]*)+$)} $resp3 ]
} then {
    set resp3 [regsub -all {(^([[:space:]]*\***)*)|((\
**[[:space:]]*)*)$)} $resp3 {}]
    set ok3 1
} else {
    set resp3 [string trim $resp3]
    set ok3 0
}

    if { [ regexp {(^([[:space:]]*\***)*)|((\**+[:space:]]*)+$)} $resp4 ]
} then {
    set resp4 [regsub -all {(^([[:space:]]*\***)*)|((\
**[[:space:]]*)*)$)} $resp4 {}]
    set ok4 1
} else {
    set resp4 [string trim $resp4]
    set ok4 0
}

    if {$mkyes} \
    then {
        dpdg eval {INSERT INTO preguntas (tema,pregunta,\
                                respuesta1,ok1,\
                                respuesta2,ok2,\
                                respuesta3,ok3,\
                                respuesta4,ok4,\
                                anulada,obsoleta,markdown)\
                                VALUES ($numTemaID,$enunciado,\
                                $resp1,$ok1,\
                                $resp2,$ok2,\

```

```

                                $resp3,$ok3,\
                                $resp4,$ok4,\
                                $anulada,$obsoleta,$mdw)
                                }
                                incr numeroPreguntasCheck
                                } \
                                else {
                                    dpdg eval {INSERT INTO preguntas (tema,pregunta,\
                                                respuesta1,ok1,\
                                                respuesta2,ok2,\
                                                respuesta3,ok3,\
                                                respuesta4,ok4,\
                                                anulada,obsoleta)\
                                                VALUES ($numTemaID,$enunciado,\
                                                $resp1,$ok1,\
                                                $resp2,$ok2,\
                                                $resp3,$ok3,\
                                                $resp4,$ok4,\
                                                $anulada,$obsoleta)
                                    }
                                    incr numeroPreguntasCheck
                                }

                                }; # fin del foreach pregtema

                                }; # fin del foreach keytema

                                if {$numeroPreguntas == $numeroPreguntasCheck} then {
                                    if {[tk_messageBox \
                                        -message "importado correctamente!" \
                                        -type ok \
                                        -parent .meterTxt] eq "ok"} {destroy .meterTxt}
                                } else {
                                    if {[tk_messageBox \
                                        -message "ha habido algún problema" \
                                        -type ok \
                                        -parent .meterTxt] eq "ok"} {return}; # aquí no sé qué pasará
                                }

                                }; # fin del button .meterTxt.flow.bimportar

                                pack .meterTxt.flow.bimportar

                                pack .meterTxt.ftop .meterTxt.flow

                                }

```

```

#####
#      FIN del procedimiento para meter txt      #
#####

```

```

#####
#### procedimiento para crear una nueva base de datos      #####
#####

```

```

proc crear {} {
    #toplevel .crear

```

```

#grab set .crear; # chequear si hace falta un grab release!

# la nueva base creada tendrá siempre la columna markdown

global tipos dpdg mkyes base

# nombre -provisional- del fichero; habrá que añadirle la extensión?
set nuevaBase [tk_getSaveFile -initialdir [pwd] -filetypes $tipos]

# sea cual sea el nombre que le pasemos le vamos a poner extensión .db
# y si había un fichero anterior con ese nombre lo copio añadiendo _(copia)
al final
# ojo con el puto windows que no distingue mayúsculas y minúsculas
set nuevaBase "[file rootname $nuevaBase].db"
set listaFich [glob -nocomplain $nuevaBase]
foreach nomFich $listaFich {file rename $nomFich "[file rootname
$nomFich]_(copia)[file extension $nomFich]}"}
# set rootNombre [file rootname $nuevaBase]
# if {[file exists "$rootNombre.db"] || [file exists "$rootNombre.DB"] ||
[file exists "$rootNombre.Db"] || [file exists "$rootNombre.dB"]} \
# then {file rename -force $nuevaBase "[file rootname $nuevaBase]_(copia)
[file extension $nuevaBase]}"}

#creo la nueva base
sqlite3 nbase $nuevaBase -create true
nbase eval {CREATE TABLE temas(id INTEGER PRIMARY KEY UNIQUE,\
                                numero INTEGER UNIQUE NOT NULL,\
                                tituloTema TEXT UNIQUE NOT NULL )}
nbase eval {CREATE TABLE preguntas(numero INTEGER PRIMARY KEY UNIQUE,\
                                    tema INTEGER NOT NULL REFERENCES temas (id),\
                                    pregunta TEXT NOT NULL,\
                                    respuesta1 TEXT NOT NULL,\
                                    ok1 INTEGER,\
                                    respuesta2 TEXT NOT NULL,\
                                    ok2 INTEGER,\
                                    respuesta3 TEXT NOT NULL,\
                                    ok3 INTEGER,\
                                    respuesta4 TEXT NOT NULL,\
                                    ok4 INTEGER,\
                                    anulada INTEGER,\
                                    obsoleta INTEGER, \
                                    markdown INTEGER )}

nbase close

if {[regexp dpdg [info commands dpdg]]} {dpdg close}
sqlite3 dpdg $nuevaBase -create false -readonly false
set base [file tail $nuevaBase]
set mkyes true
if {[regexp dpdg [info commands dpdg]]} then {
    if {[tk_messageBox \
        -message "Base creada y seleccionada!" \
        -type ok \
        -parent .] eq "ok"} {return}
} else {
    if {[tk_messageBox \
        -message "Ups, ha habido algún problema!" \
        -type ok \
        -parent .] eq "ok"} {return}
}
}

#####
#          fin del procedimiento crear          #####

```

```
#####
```

```
#####  
#### procedimiento para gestionar temas #####  
#####
```

```
proc temas {} {  
    # se ejecuta al pulsar el botón "Gestionar temas"  
    # supongo que habrá que comprobar que hay una base de datos válida  
    seleccionada  
    # en principio asumo que sí  
  
    if {[regexp dpdg [info commands dpdg]]} {  
        tk_messageBox -message "no hay una base de datos seleccionada" -type ok  
        return  
    }  
  
    toplevel .temas  
    grab set .temas; # chequear si hace falta un grab release!  
  
    global dpdg ruta_base cfond cbad ii listaii cambiosOK nuevOTEMA  
  
    set listaii {}  
  
    frame .temas.ftitulos  
    frame .temas.fcontenidos  
    frame .temas.fnuevotema -bd 1 -pady 16 -highlightcolor green  
    frame .temas.fbotones  
    label .temas.ftitulos.l1 -text "id" -height 3 -width 6  
    label .temas.ftitulos.l2 -text "Nº Tema" -height 3 -width 8  
    label .temas.ftitulos.l3 -text "Título Tema" -height 3 -width 40  
    label .temas.ftitulos.l4 -text "Preguntas Válidas" -height 3 -width 18  
    label .temas.ftitulos.l5 -text "Preguntas Anuladas" -height 3 -width 18  
    label .temas.ftitulos.l6 -text "Preguntas Obsoletas" -height 3 -width 18  
    # puts $ruta_base  
  
    # set ii 0  
    # voy a hacer que ii sea igual que id  
    # me creo una lista con los ii (o id, que es lo mismo))  
  
    dpdg eval "SELECT id,numero,tituloTema FROM temas ORDER BY numero"  
    tablaTemas {  
        # incr ii  
        set ii $tablaTemas(id)  
        lappend listaii $ii  
        set l4 [dpdg eval "SELECT count(*) FROM preguntas WHERE tema =  
$tablaTemas(id) AND (NOT anulada) AND (NOT obsoleta)"]  
        set l5 [dpdg eval "SELECT count(*) FROM preguntas WHERE tema =  
$tablaTemas(id) AND anulada"]  
        set l6 [dpdg eval "SELECT count(*) FROM preguntas WHERE tema =  
$tablaTemas(id) AND obsoleta"]  
        frame .temas.fcontenidos.f$i  
        label .temas.fcontenidos.f$i.ll1 -text $tablaTemas(id) -height 1 -width 6  
        set ::num$i $tablaTemas(numero)  
        set ::tit$i $tablaTemas(tituloTema)  
        entry .temas.fcontenidos.f$i.ll2 -textvariable ::num$i -width 8 -bg  
white -justify center \  
        -vcmd {if {[string is integer -strict %P] && [expr {%P > 0}]} {%W  
configure -background $cfond} {%W configure -bg $cbad}; return 1} \  
        -validate none  
        entry .temas.fcontenidos.f$i.ll3 -textvariable ::tit$i -width 40 -bg
```

```

white -justify left \
    -vcmd {if {[string match -nocase {[a-z]*} %P]} {%W configure -
background $cfond} {%W configure -bg $cbad}; return 1} \
    -validate none
    label .temas.fcontenidos.f$ii.ll4 -text "$l4" -height 1 -width 18
    label .temas.fcontenidos.f$ii.ll5 -text "$l5" -height 1 -width 18
    label .temas.fcontenidos.f$ii.ll6 -text "$l6" -height 1 -width 18

pack .temas.fcontenidos.f$ii.ll1 .temas.fcontenidos.f$ii.ll2 .temas.fcontenidos.
f$ii.ll3 \
    .temas.fcontenidos.f$ii.ll4 .temas.fcontenidos.f$ii.ll5 .temas.fconten
idos.f$ii.ll6 -side left
    pack .temas.fcontenidos.f$ii
    .temas.fcontenidos.f$ii.ll2 configure -validate key
    .temas.fcontenidos.f$ii.ll3 configure -validate key
}

#####

button .temas.fbotones.guardar -text "Guardar Cambios" -command {
    # hay que chequear que todo está ok;
    #lo controlo con la variable booleana cambiosOK
    set cambiosOK true
    set nuevoTEMA false
    set listnum {}
    set listtit {}

    foreach li $listaii {
        lappend listnum [set ::num$li]
        lappend listtit [set ::tit$li]
    }

    foreach li $listnum {
        if {[![string is integer -strict $li] && [expr {$li > 0}]]} {set
cambiosOK false}; # número de tema no válido
    }

    foreach li $listtit {
        if {[![string match -nocase {[a-z]*} $li]]} {set cambiosOK false}; #
título de tema no válido
    }

    if {[expr {[llength [lsort $listnum]] != [llength [lsort -unique
$listnum]]}] } {set cambiosOK false}; # num tema repetido
    if {[llength [lsort $listtit]] != [llength [lsort -unique $listtit]]} {set
cambiosOK false}; # tit tema repetido
    if {[tcl::mathop::in $::numnt $listnum]} {set cambiosOK false}; # num del
nuevo tema repetido
    if {[tcl::mathop::in $::titnt $listtit]} {set cambiosOK false}; # tit del
nuevo tema repetido
    if {[![string is space $::numnt] || ![string is space $::titnt]]} {
        set nuevoTEMA true
        if {[![string is integer -strict $::numnt] && [expr {$::numnt > 0}]]}
{set cambiosOK false}; # num del nuevo tema no válido
        if {[![string match -nocase {[a-z]*} $::titnt]]} {set cambiosOK
false}; # tit del nuevo tema no válido
    }
    # puts $cambiosOK
    if {$cambiosOK} {
        # escribo en la base de datos usando una transacción
        # al terminar habrá que recargar el widget con la nueva tabla
        dpdg transaction {
            foreach li $listaii {

```

```

        dpdg eval "UPDATE temas SET numero=[set ::num$li],
tituloTema='[set ::tit$li]' WHERE id=$li"
    }
    if {$nuevoTEMA} {
        dpdg eval "INSERT INTO temas (numero,tituloTema) VALUES
($::numnt,'$::titnt')"
    }
}
# temas (esto no furrula))
# return (esto tampoco)
set ::numnt {}
set ::titnt {}
after idle temas
destroy .temas
}
}

button .temas.fbotones.cerrar -text Cerrar -command {
    destroy .temas
    set ::numnt {}
    set ::titnt {}
}

label .temas.fnuevotema.ll1nt -text "Nuevo\nTema" -height 2 -width 6

#####

entry .temas.fnuevotema.ll2nt -textvariable ::numnt -width 6 -bg white -
justify center -font {Helvetica -18 bold} \
    -vcmd {
        if {[string is integer -strict %P] && [expr {%P > 0}]} then
        {.temas.fnuevotema.ll2nt configure -background $cfond; if {[string equal
$::titnt ""]} {.temas.fnuevotema.ll3nt configure -bg $cbad}} \
            elseif {[string equal %P ""]} && [string equal $::titnt ""]}
        {.temas.fnuevotema.ll2nt configure -bg white; .temas.fnuevotema.ll3nt configure
-background white} \
            else {.temas.fnuevotema.ll2nt configure -bg $cbad}
        return 1} \
    -validate key

#####

entry .temas.fnuevotema.ll3nt -textvariable ::titnt -width 40 -bg white -
justify left -font {Helvetica -18 bold} \
    -vcmd {
        if {[string match -nocase {[a-z]*} %P]} then {.temas.fnuevotema.ll3nt
configure -background $cfond; if {[string equal $::numnt ""]}
        {.temas.fnuevotema.ll2nt configure -bg $cbad}} \
            elseif {[string equal %P ""]} && [string equal $::numnt ""]}
        {.temas.fnuevotema.ll3nt configure -background white; .temas.fnuevotema.ll2nt
configure -bg white} \
            else {.temas.fnuevotema.ll3nt configure -bg $cbad}
        return 1} \
    -validate key

#####

label .temas.fnuevotema.ll4nt -text "" -height 2 -width 8
label .temas.fnuevotema.ll5nt -text "" -height 2 -width 18
label .temas.fnuevotema.ll6nt -text "" -height 2 -width 18

pack .temas.ftitulos.l1 .temas.ftitulos.l2 .temas.ftitulos.l3 \
    .temas.ftitulos.l4 .temas.ftitulos.l5 .temas.ftitulos.l6 -side left
pack .temas.fnuevotema.ll1nt .temas.fnuevotema.ll2nt .temas.fnuevotema.ll3nt

```



```

\
    .temas.fnuevotema.ll4nt .temas.fnuevotema.ll5nt .temas.fnuevotema.ll6nt -
side left
    pack .temas.fbotones.guardar .temas.fbotones.cerrar -side left
    pack .temas.ftitulos .temas.fcontenidos .temas.fnuevotema .temas.fbotones
}

pack .fsup .falt .fmed .fmedbajo .finf

##### fin procedimiento temas #####

#####
#### procedimiento para generar gifts #####
#####

proc generar {} {
    if {[regexp dpdg [info commands dpdg]]} {
        tk_messageBox -message "no hay una base de datos seleccionada" -type ok
        return
    }

    toplevel .generar
    grab set .generar
    wm attributes .generar -topmost true

    global mkyes

    # creo una frame que contendrá una label con el mensaje de bienvenida
    # ojo que el procedimiento es generar y la toplevel es .generar
    frame .generar.fwelcome
    label .generar.fwelcome.lwelcome1 -text "GENERAR ARCHIVO .GIFT PARA MOODLE"
    label .generar.fwelcome.lwelcome2 -text "indique los temas que se
incluirán:"

    pack .generar.fwelcome.lwelcome1 .generar.fwelcome.lwelcome2
    pack .generar.fwelcome

    # set ::respuestaMal "~%-25%"
    set ::respuestaMal "~%-33.33333%"
    set ::respuestaBiende1 "="
    set ::respuestaBiende2 "~%50%"
    set ::respuestaBiende3 "~%33.33333%"
    set ::respuestaBiende4 "~%25%"
    # set ::respuesta5 "\t~%0%\[moodle\]No se selecciona ninguna respuesta."
    set ::respuesta5 "\t~%0%No se selecciona ninguna respuesta."

    # num_usuario debería ser una lista de temas seleccionados
    # tengo que montar algo con checkbuttons para seleccionarlos
    # set numeros_tema_usuario {2 4}
    set ::todos_numeros_tema_usuario [dpdg eval {SELECT numero FROM temas}]
    set ::todos_numeros_tema_usuario [lsort -integer
$::todos_numeros_tema_usuario]
    # nt será el número de temas que hay
    # nf será el número de frames que uso para meter los checkbuttons
    # teniendo en cuenta que meto 3 checkbuttons por frame (salvo quizás en la
última)
    set ::nt [llength $::todos_numeros_tema_usuario]
    set ::nf [expr int(ceil($::nt/3))]
    set ::rf [expr $::nt%3]

    set metidos_total 0
    for {set ift 1} {[expr "$ift <= $::nf"]} {incr ift} {

```

```

frame .generar.ftemas$ift
set metidos 0
while {[expr ($metidos < 3) && ($metidos_total < $::nt)]} {
    incr metidos
    incr metidos_total
    checkbutton .generar.ftemas$ift.buttonnt$metidos_total -text "tema
[linde $::todos_numeros_tema_usuario [expr $metidos_total - 1]]" -variable
tt$metidos_total
    pack .generar.ftemas$ift.buttonnt$metidos_total -side left
}

pack .generar.ftemas$ift
}

```

```

if {$::rf > 0} {
    set frameResto [expr $::nf +1]
    puts $frameResto
    frame .generar.ftemas$frameResto
    set metidos 0
    while {[expr ($metidos < 3) && ($metidos_total < $::nt)]} {
        incr metidos
        incr metidos_total; puts "met_total $metidos_total"
        checkbutton .generar.ftemas$frameResto.buttonnt$metidos_total -text
"tema [linde $::todos_numeros_tema_usuario [expr $metidos_total - 1]]" -
variable tt$metidos_total
        pack .generar.ftemas$frameResto.buttonnt$metidos_total -side left
    }
    pack .generar.ftemas$frameResto;
}

```

#####

```

frame .generar.fpregplus
frame .generar.fpregminus

label .generar.fpregplus.lplus -text "Preguntas a añadir: "
entry .generar.fpregplus.eplus -textvariable preguntasmas -width 20 -bg
OliveDrab1 -font bold

label .generar.fpregminus.lminus -text "Preguntas a quitar: "
entry .generar.fpregminus.eminus -textvariable preguntasmenos -width 20 -bg
salmon -font bold

```

```

pack .generar.fpregplus.lplus .generar.fpregplus.eplus -side left
pack .generar.fpregminus.lminus .generar.fpregminus.eminus -side left

```

```

frame .generar.fporcenresta
frame .generar.fporcenresta.fizqui
frame .generar.fporcenresta.fderech
label .generar.fporcenresta.fizqui.lporcentaje -text "% que restan las
erróneas: "
ttk::radiobutton .generar.fporcenresta.fderech.rb25 -text "25.0%" -
variable ::respuestaMal -value "~%-25%"
ttk::radiobutton .generar.fporcenresta.fderech.rb33 -text "33.3%" -
variable ::respuestaMal -value "~%-33.33333%"

```

```

pack .generar.fporcenresta.fderech.rb25 .generar.fporcenresta.fderech.rb33
pack .generar.fporcenresta.fizqui.lporcentaje
pack .generar.fporcenresta.fizqui .generar.fporcenresta.fderech -side left

```

```

frame .generar.fpregunta5
label .generar.fpregunta5.lpregunta5 -text "¿Incluir quinta pregunta?: "
checkbutton .generar.fpregunta5.cbpregunta5 -text "Actualmente No. Marcar \

```

```

n para cambiar" -state normal -variable p5 -command {
    if {$p5} {.generar.fpregunta5.cbpregunta5 configure -text "Actualmente Sí.
Desmarcar \n para cambiar"} {.generar.fpregunta5.cbpregunta5 configure -text
"Actualmente No. Marcar \n para cambiar "}
}

pack .generar.fpregunta5.lpregunta5 .generar.fpregunta5.cbpregunta5 -side
left

frame .generar.fbgenerar

button .generar.fbgenerar.bgenerar -text "Generar Fichero" -command {

    set numero_de_preguntas 0
    set pp5 $p5
    if {$pp5} {set nomCon5 "_con_5_resp"} {set nomCon5 ""}
    set numeros_tema_usuario_prov {}
    for {set mm 1} {[expr $mm <= $::nt]} {incr mm} {
        if {[set tt$mm]} {lappend numeros_tema_usuario_prov [lindex
$::todos_numeros_tema_usuario [expr $mm -1]]}
    }
    set ladd {}
    foreach la $preguntasmas {
        if {[expr [string is integer -strict $la]]} { if {[expr ($la > 0)]}
{lappend ladd $la } }
    }
    set ladd [lsort -integer $ladd]
    set lrest $ladd
    foreach lr $preguntasmenos {
        if {[expr [string is integer -strict $lr]]} { if {[expr ($lr > 0)]}
{lappend lrest $lr } }
    }
    # set ladd "([string map {{ } ,} $ladd])"
    if {[llength $lrest]} {set cambios "&"} {set cambios ""}
    set lrest "([string map {{ } ,} $lrest])"
    puts $ladd
    puts $lrest
    set numeros_tema_usuario $numeros_tema_usuario_prov
    if {[llength $numeros_tema_usuario] >1} {set plural "s_"} {set plural "_"}

    set nombre_fichero [string map {{ } _} $numeros_tema_usuario]
    set salida [open tema$plural$nombre_fichero$cambios$nomCon5.txt w]

    foreach num_usuario $numeros_tema_usuario {
        set num_id [dpg eval {SELECT id FROM temas WHERE numero =
$num_usuario}]
        if {$mkyes} {set tramoMK ",markdown"} {set tramoMK ""}
        dpg eval "SELECT numero, tema, pregunta, respuesta1, ok1, respuesta2,
ok2, respuesta3, ok3, respuesta4, ok4 $tramoMK FROM preguntas WHERE tema =
$num_id AND (NOT anulada) AND (NOT obsoleta) AND (numero NOT IN $lrest)"
valores {
            #      incr ii
            set totalOK [expr $valores(ok1)+$valores(ok2)+$valores(ok3)+
$valores(ok4)]
            switch $totalOK {
                1 {set respuestaBien $::respuestaBiende1}
                2 {set respuestaBien $::respuestaBiende2}
                3 {set respuestaBien $::respuestaBiende3}
                4 {set respuestaBien $::respuestaBiende4}
            }

            set valores(pregunta) [regsub -all {(:|#|{|}|~|=)}
$valores(pregunta) {\\1}]
            set valores(respuesta1) [regsub -all {(:|#|{|}|~|=)}

```

```

$valores(respuesta1) {\#\1}
    set valores(respuesta2) [regsub -all {(:|#|{|}|\~|=)}
$valores(respuesta2) {\#\1}
    set valores(respuesta3) [regsub -all {(:|#|{|}|\~|=)}
$valores(respuesta3) {\#\1}
    set valores(respuesta4) [regsub -all {(:|#|{|}|\~|=)}
$valores(respuesta4) {\#\1}

    set antepre "\[html\]<p><strong>"
    set postpre "</strong></p>"
    if {$mkyes} { if {$valores(markdown)} {set antepre "\[markdown\]";
set postpre "" } }

    puts $salida "::$T[format %02d $num_usuario]_P[format %03d
$valores(numero)]:$antepre$valores(pregunta)$postpre{"
    puts $salida "\t[if {$valores(ok1)} {set respuestaBien}
{set ::respuestaMal}]$valores(respuesta1)"
    puts $salida "\t[if {$valores(ok2)} {set respuestaBien}
{set ::respuestaMal}]$valores(respuesta2)"
    puts $salida "\t[if {$valores(ok3)} {set respuestaBien}
{set ::respuestaMal}]$valores(respuesta3)"
    puts $salida "\t[if {$valores(ok4)} {set respuestaBien}
{set ::respuestaMal}]$valores(respuesta4)"
    if {$pp5} { puts $salida $::respuesta5 }
    puts $salida "}"
    puts $salida ""
    incr numero_de_preguntas
}
}

foreach llaa $ladd {
    if {$mkyes} {set tramoMK ",markdown"} {set tramoMK ""}
    dpdg eval "SELECT numero, tema, pregunta, respuesta1, ok1, respuesta2,
ok2, respuesta3, ok3, respuesta4, ok4 $tramoMK FROM preguntas WHERE numero =
$llaa" valores {
        set num_tem_us [dpdg eval {SELECT numero FROM temas WHERE id =
$valores(tema)}]
        #      incr ii
        set totalOK [expr $valores(ok1)+$valores(ok2)+$valores(ok3)+
$valores(ok4)]
        switch $totalOK {
            1 {set respuestaBien $::respuestaBiende1}
            2 {set respuestaBien $::respuestaBiende2}
            3 {set respuestaBien $::respuestaBiende3}
            4 {set respuestaBien $::respuestaBiende4}
        }

        set valores(pregunta) [regsub -all {(:|#|{|}|\~|=)}
$valores(pregunta) {\#\1}
        set valores(respuesta1) [regsub -all {(:|#|{|}|\~|=)}
$valores(respuesta1) {\#\1}
        set valores(respuesta2) [regsub -all {(:|#|{|}|\~|=)}
$valores(respuesta2) {\#\1}
        set valores(respuesta3) [regsub -all {(:|#|{|}|\~|=)}
$valores(respuesta3) {\#\1}
        set valores(respuesta4) [regsub -all {(:|#|{|}|\~|=)}
$valores(respuesta4) {\#\1}

        set antepre "\[html\]<p><strong>"
        set postpre "</strong></p>"
        if {$mkyes} { if {$valores(markdown)} {set antepre "\[markdown\]";
set postpre "" } }

        puts $salida "::$T[format %02d $num_tem_us]_P[format %03d

```

```

$valores(numero)]::$antepre$valores(pregunta)$postpre{"
    puts $salida "\t[if {$valores(ok1)} {set respuestaBien}
{set ::respuestaMal}]$valores(respuesta1)"
    puts $salida "\t[if {$valores(ok2)} {set respuestaBien}
{set ::respuestaMal}]$valores(respuesta2)"
    puts $salida "\t[if {$valores(ok3)} {set respuestaBien}
{set ::respuestaMal}]$valores(respuesta3)"
    puts $salida "\t[if {$valores(ok4)} {set respuestaBien}
{set ::respuestaMal}]$valores(respuesta4)"
    if {$pp5} { puts $salida "$::respuesta5" }
    puts $salida "}"
    puts $salida ""
    incr numero_de_preguntas
}
}
# aquí cierro el fichero que he generado
close $salida
if {[tk_messageBox \
    -message "Generado archivo
tema$plural$nombre_fichero$cambios$nomCon5.txt \n con $numero_de_preguntas
preguntas" \
    -type ok \
    -parent .generar] eq "ok"} {return}

}

pack .generar.fbgenerar.bgenerar

pack .generar.fpregminus
pack .generar.fpregplus
pack .generar.fporcenresta
pack .generar.fpregunta5
pack .generar.fbgenerar

}

#### fin procedimiento para generar gifts #####

#####
#### procedimiento para mostrar preguntas #####
#####

proc mostrar {} {
    if {[regexp dpdg [info commands dpdg]]} then {
        if {[dpdg eval {SELECT COUNT(*) FROM temas}]<1} {
            tk_messageBox -message "la base de datos no tiene ningún tema" -type
ok
            return
        }
    } else {
        tk_messageBox -message "no hay una base de datos seleccionada" -type ok
        return
    }
}

# el procedimiento es mostrar y la toplevel es .mostrar
toplevel .mostrar
grab set .mostrar

global preg

#####
# creo una frame con una label que contiene un encabezamiento general

```

```
#####

frame .mostrar.fwelcome
label .mostrar.fwelcome.lwelcome1 -text "MOSTRAR PREGUNTAS DE TEST POR
TEMAS"

pack .mostrar.fwelcome.lwelcome1

#####
# creo una frame para meter dos botones, para mostrar y para exportar
#####
frame .mostrar.fbotones
button .mostrar.fbotones.bmostrar -text "Mostrar Preguntas" -command {
    set nTema [.mostrar.ftemas.clistatemas get]
    if {[dict exists $::dicIdTemas $nTema]} {
        set idTema [dict get $::dicIdTemas $nTema]
        set numTema [lindex [dict filter $::dicNumTemas value $nTema] 0]
        mmostrar $idTema $nTema $numTema
    }
}
button .mostrar.fbotones.bexportar -text "Exportar a txt" -command {} -state
disabled

pack .mostrar.fbotones.bmostrar .mostrar.fbotones.bexportar -side left

# aquí voy a guardar la lista de temas de la base de datos
# para ello creo un diccionario para almacenar el número real (la id) de
cada tema
# y su título (a partir de la tabla temas): dicIdTemas
# también creo una lista con los títulos de los temas para el combobox
# estos títulos los ordeno de acuerdo con el número de los temas
# para ello casi mejor me creo otro diccionario con el número y el nombre de
cada tema: dicNumTemas

set ::dicIdTemas [dict create]
set ::dicNumTemas [dict create]
lappend ::listaTemas

dpdg eval {SELECT id,numero,tituloTema FROM temas} valores {
    dict append ::dicIdTemas $valores(tituloTema) $valores(id)
    dict append ::dicNumTemas $valores(numero) $valores(tituloTema)
}

set ::dicNumTemas [lsort -integer -stride 2 $::dicNumTemas]
set ::listaTemas [dict values $::dicNumTemas]
# puts $::listaTemas
# puts [llength $::listaTemas]
set 1nTema [lindex $::listaTemas 0]
set 1numTema [lindex $::dicNumTemas 0]
set 1id [lindex [dict filter $::dicIdTemas key $1nTema] 1]

#####
# creo una frame para la lista de temas y los checkbuttons de
anuladas/obsoletas

#####
frame .mostrar.ftemas
ttk::combobox .mostrar.ftemas.clistatemas -width 40 -justify left -state
normal -values $::listaTemas -textvariable ::nombreTema
checkbox .mostrar.ftemas.canuladas -text "Mostrar Anuladas" -state normal
-variable mAnuladas
checkbox .mostrar.ftemas.cobsoletas -text "Mostrar Obsoletas" -state
normal -variable mObsoletas
```

```
pack .mostrar.ftemas.clistatemas .mostrar.ftemas.canuladas .mostrar.ftemas.cobso  
letas -side left
```

```
#####  
# me creo una frame para el widget text y la barra de desplazamiento  
#####  
frame .mostrar.fpreguntas  
set preg [text .mostrar.fpreguntas.tpreguntas -background #ffffff -wrap word  
-undo 1 -yscrollcommand [list .mostrar.fpreguntas.sbarra set]]  
set barra [ttk::scrollbar .mostrar.fpreguntas.sbarra -command [list $preg  
yview]]  
pack .mostrar.fpreguntas.tpreguntas -side left -fill both -expand true  
pack .mostrar.fpreguntas.sbarra -side left -fill y
```

```
set ::nombreTema $1nTema  
mmostrar $1id $1nTema $1numTema
```

```
pack .mostrar.fwelcome  
pack .mostrar.fbotones  
pack .mostrar.ftemas  
pack .mostrar.fpreguntas -expand true -fill both
```

```
}
```

```
#### fin procedimiento para mostrar preguntas #####
```

```
# me crearé un procedimiento que es el que se ejecutará al pulsar al botón  
mostrar
```

```
proc mmostrar {id nn numnum} {  
    global preg  
    # global nTema numTema  
    global mAnuladas mObsoletas mkyes  
    set ansOK "\{ansOK\}"  
    set ansOK1 " ansOK"  
    $preg delete 1.0 end  
    $preg tag configure titulo -font {times 18 bold} -justify center  
    $preg tag configure bold -font {times 12 bold}  
    $preg tag configure centro -justify center -font {times 14 bold}  
    $preg tag configure anulada -foreground red  
    set formatPreg "bold"  
    $preg tag configure ansOK -underline true  
    $preg tag configure margenpreg -lmargin1 10 -lmargin2 50 -rmargin 10  
    $preg tag configure margenresp1 -lmargin1 90 -rmargin 10  
    $preg tag configure margenresp2 -lmargin2 110 -rmargin 10  
    $preg insert end "\n"  
    $preg insert end "Tema $numnum. $nn" {titulo}  
    $preg insert end "\n"  
    $preg insert end "\{(Las respuestas correctas están subrayadas\)" {centro}  
    $preg insert end "\n"  
    $preg insert end "\{(Las preguntas anuladas/obsoletas están en rojo\)"  
{anulada centro}  
    $preg insert end "\n\n"  
    set filtro "tema = $id"  
    if {[expr ! $mAnuladas]} {append filtro " AND NOT anulada"}  
    if {[expr ! $mObsoletas]} {append filtro " AND NOT obsoleta"}  
    if {$mkyes} {set tramoMK " ,markdown"} {set tramoMK ""}  
    set consulta "SELECT  
pregunta, numero, respuesta1, respuesta2, respuesta3, respuesta4, \  
ok1, ok2, ok3, ok4, anulada, obsoleta $tramoMK FROM preguntas  
WHERE $filtro"  
    # puts $consulta
```

```

dpdg eval $consulta textoPreg {
    if {[expr $textoPreg(anulada) || $textoPreg(obsoleta)]} {set formatPreg
"anulada"} {set formatPreg "bold"}
    # puts $textpPreg(pregunta)
    set flagMK ""
    if {$mkyes} { if {$textoPreg(markdown)} {set flagMK "\[mk\]"} }
    $preg insert end " P-$textoPreg(numero)\) $flagMK$textoPreg(pregunta)\n"
"$formatPreg margenpreg"
    $preg insert end "1\)" {margenresp1};
    $preg insert end "$textoPreg(respuesta1)\n" "margenresp1 margenresp2 [if
{$textoPreg(ok1)} {set ansOK1}]"
    $preg insert end "2\)" {margenresp1};
    $preg insert end "$textoPreg(respuesta2)\n" "margenresp1 margenresp2 [if
{$textoPreg(ok2)} {set ansOK1}]"
    $preg insert end "3\)" {margenresp1};
    $preg insert end "$textoPreg(respuesta3)\n" "margenresp1 margenresp2 [if
{$textoPreg(ok3)} {set ansOK1}]"
    $preg insert end "4\)" {margenresp1};
    $preg insert end "$textoPreg(respuesta4)\n" "margenresp1 margenresp2 [if
{$textoPreg(ok4)} {set ansOK1}]"
    $preg insert end "\n"
}
puts 1
}

```

```

#####
#### procedimiento para editar preguntas #####
#####

```

```

proc editar {} {
    # global dpdg

    if {[regexp dpdg [info commands dpdg]]} then {
        if {[dpdg eval {SELECT COUNT(*) FROM temas}]<1} {
            tk_messageBox -message "la base de datos no tiene ningún tema" -type
ok
            return
        }
    } else {
        tk_messageBox -message "no hay una base de datos seleccionada" -type ok
        return
    }

    # el procedimiento es editar y la toplevel es .editar
    toplevel .editar
    grab set .editar

    global maxfilas numtemas nuevapreg preg npflag guardar
    global numero enunciado respuesta1 respuesta2 respuesta3 respuesta4
nombreTema preg lectura
    global cb_ok1 cb_ok2 cb_ok3 cb_ok4 anulada obsoleta markdown

    # creo un diccionario para almacenar el número real (la id) de cada tema
    # y su título (a partir de la tabla temas)
    # tambien creo una lista con los títulos de los temas para el combobox

    set maxfilas [dpdg eval {SELECT count(*) FROM preguntas}]
    set numtemas [dpdg eval {SELECT count(*) FROM temas}]

    set ::dicIdTemas [dict create]
    set ::dicIdTemasRev [dict create]
    set ::dicNumTemas [dict create]
    lappend ::listaTemas

```



```

dpdg eval {SELECT id,numero,tituloTema FROM temas} valores {
    dict append ::dicIdTemas $valores(tituloTema) $valores(id)
    dict append ::dicIdTemasRev $valores(id) $valores(tituloTema)
    dict append ::dicNumTemas $valores(numero) $valores(tituloTema)
}

set ::dicNumTemas [lsort -integer -stride 2 $::dicNumTemas]
set ::listaTemas [dict values $::dicNumTemas]

# enunciado contendrá el enunciado de la pregunta
# numero contendrá el número de la pregunta

set numero 1
set enunciado {}
set respuesta1 {}
set respuesta2 {}
set respuesta3 {}
set respuesta4 {}
set nombreTema {}
set cb_ok1 0
set cb_ok2 0
set cb_ok3 0
set cb_ok4 0
set anulada 0
set obsoleta 0
set markdown 0

# lectura es un flag que indica si se acaba de leer la base de datos
# es decir, se pone a 1 al ejecutar p.leerfilas
set lectura 0

# npflag es un flag que indica si se está introduciendo una nueva pregunta
en el formulario
# es decir, una nueva fila que habrá que guardar en la base de datos
set npflag 0

##### estructura gráfica con 7 frames

frame .editar.fpregunta
frame .editar.frespuesta1
frame .editar.frespuesta2
frame .editar.frespuesta3
frame .editar.frespuesta4
frame .editar.ftitulo
frame .editar.fnumero

#### en la fpregunta meto una label y un text
label .editar.fpregunta.lpregunta -text "Pregunta:"
set preg [text .editar.fpregunta.tenunciado -width 100 -height 8 -
background #ffffff -wrap word -undo 1]
pack .editar.fpregunta.lpregunta .editar.fpregunta.tenunciado -side left

#### en las frespuesta meto una label y una entry y un checkbutton
foreach i {1 2 3 4} {
    label .editar.frespuesta$i.lrespuesta$i -text "Respuesta $i:"
    entry .editar.frespuesta$i.erespuesta$i -textvariable "respuesta$i" -
width 100
    checkbutton .editar.frespuesta$i.cb_ok$i -text "ok"

pack .editar.frespuesta$i.lrespuesta$i .editar.frespuesta$i.erespuesta$i .editar
.frespuesta$i.cb_ok$i -side left
}

```

```

##### en la ftitulo meto una label y un combobox y 2 checkbuttons de anulada
y obsoleta (y otro de markdown)
label .editar.ftitulo.ltitulo -text "Tema: "
ttk::combobox .editar.ftitulo.ctitulo -width 40 -justify left -state normal
-values $::listaTemas -textvariable nombreTema
checkboxbutton .editar.ftitulo.cb_anulada -text "anulada" -variable anulada
checkboxbutton .editar.ftitulo.cb_obsoleta -text "obsoleta" -variable obsoleta
checkboxbutton .editar.ftitulo.cb_markdown -text "markdown" -variable markdown
-state disabled

pack .editar.ftitulo.ltitulo .editar.ftitulo.ctitulo .editar.ftitulo.cb_anulada
.editar.ftitulo.cb_obsoleta .editar.ftitulo.cb_markdown -side left

##### en la .fnumero meto una label y una entry y dos buttons de incrementar
y decrementar
##### meto también el botón de guardar cambios, con una label y un botón
##### meto también el botón de nueva pregunta
label .editar.fnumero.lnumero -text "Num:"
entry .editar.fnumero.ennumero -textvariable numero -width 10 -justify center
-validate key \
    -vcmd {if {[string is digit -strict %P]}\
        then {if { [expr %P <= ($maxfilas +1)] && [expr %P > 0]}\
            then {after idle {desactivar ; p.leerfilas %P;
activar}};return -level 2 1}\
            else {return -level 2 0}}\
        else {if {%P eq ""}\
            then {after idle { desactivar ; p.leerfilas %P;
activar }};return -level 2 1}\
            else {return -level 2 0}}\
        }
button .editar.fnumero.bincrementar -width 4 -text "+" \
    -command {if {[string is digit $numero]}\
        then {if {$numero eq ""}\
            then {set numero 1}\
            else {if {$numero < $maxfilas} then {incr numero}}\
        }\
    }
button .editar.fnumero.bdecrementar -width 4 -text "-" \
    -command {if {[string is digit -strict $numero]}\
        then {if {$numero > 1} then {incr numero -1}}
        puts $numero
        puts [.editar.fnumero.ennumero cget -validate]
    }

label .editar.fnumero.lguardarcambios -text "    Guardar Cambios:  "
set guardar [button .editar.fnumero.bguardar -width 6 -text "guardar" -state
disabled -command p.guardar]

label .editar.fnumero.labelnula -text "        "
set nuevapreg [button .editar.fnumero.nuevapreg -width 10 -text "nueva
pregunta" \
    -state active -activeforeground "green" -command p.nueva]

# ejecuto el procedimiento para dar valores iniciales a todo el formulario
desactivar
set numero 1
activar

bind $preg <<Modified>> {
    if {$lectura}\
        then {set lectura 0; if {[ $preg edit modified]} then { $preg edit
modified 0}}\
        else {if {[ $preg edit modified]} then { $guardar configure -state

```

```

active -activeforeground "blue"}}
}

pack .editar.fnumero.lnumero .editar.fnumero.ennumero .editar.fnumero.bincrementa
r .editar.fnumero.bdecrementar .editar.fnumero.lguardarcambios .editar.fnumero.b
guardar .editar.fnumero.labelnula .editar.fnumero.nuevapreg -side left

pack .editar.fpregunta .editar.frespuesta1 .editar.frespuesta2 .editar.frespuest
a3 .editar.frespuesta4 .editar.ftitulo .editar.fnumero

## voy a meter código para que al cerrar la ventana con la toplevel .editar
se desactiven las trazas
wm protocol .editar WM_DELETE_WINDOW {
    if {[tk_messageBox \
        -icon question \
        -type yesno \
        -default no \
        -parent .editar \
        -message "De verdad quieres salir?" \
        -title "Cerrar Aplicación"] == "yes"} {
        desactivar
        destroy .editar
    }
}

}

# aquí meto lo de trace, el procedimiento para activar el botón guardar
# antes lo tenía dentro del procedimiento editar() -espero que así esté mejor-
proc tracer {nomVarTrac index ops} {
    global guardar
    $guardar configure -state active -activeforeground "blue"
    #puts [.editar.fnumero.bguardar cget -state]
}

#### fin procedimiento para editar preguntas #####

# este procedimiento se ejecuta cada vez que cambiemos el número de pregunta
# en el formulario
# sirve para actualizar los distintos campos del formulario
proc p.leerfilas numfila {
    global enunciado respuesta1 respuesta2 respuesta3 respuesta4 nombreTema preg
lectura
    global cb_ok1 cb_ok2 cb_ok3 cb_ok4 anulada obsoleta markdown mkyes
    global maxfilas
    global nuevapreg npflag

    set markdown 0
    set lectura 1

    if { ($numfila eq "") || ($numfila == $maxfilas + 1) }\
    then {
        set enunciado {}
        set respuesta1 {}
        set respuesta2 {}
        set respuesta3 {}
        set respuesta4 {}
        set nombreTema [lindex $::listaTemas 0]
        set cb_ok1 0
        set cb_ok2 0
        set cb_ok3 0
        set cb_ok4 0
    }
}

```

```

        set anulada 0
        set obsoleta 0
        # si vamos por esta rama markdown sigue siendo 0
        # pero tengo que activar el checkbox correspondiente
        .editar.ftitulo.cb_markdown configure -state normal
        $preg replace 1.0 end $enunciado
        # lo siguiente está a prueba
        if {$maxfilas == 0} {$nuevapreg configure -state disabled; set npflag
1}
    } \
    else {
        set npflag 0
        $nuevapreg configure -state active
        set enunciadoi [dpdg eval {SELECT pregunta FROM preguntas WHERE
numero=$numfila}]
        set enunciado [string trim $enunciadoi "{}"]
        set respuesta1i [dpdg eval {SELECT respuesta1 FROM preguntas WHERE
numero=$numfila}]
        set respuesta1 [string trim $respuesta1i "{}"]
        set respuesta2i [dpdg eval {SELECT respuesta2 FROM preguntas WHERE
numero=$numfila}]
        set respuesta2 [string trim $respuesta2i "{}"]
        set respuesta3i [dpdg eval {SELECT respuesta3 FROM preguntas WHERE
numero=$numfila}]
        set respuesta3 [string trim $respuesta3i "{}"]
        set respuesta4i [dpdg eval {SELECT respuesta4 FROM preguntas WHERE
numero=$numfila}]
        set respuesta4 [string trim $respuesta4i "{}"]
        set numeroTema [dpdg eval {SELECT tema FROM preguntas WHERE
numero=$numfila}]
        set nombreTema [dict get $::dicIdTemasRev $numeroTema]
        set cb_ok1 [dpdg eval {SELECT ok1 FROM preguntas WHERE
numero=$numfila}]
        set cb_ok2 [dpdg eval {SELECT ok2 FROM preguntas WHERE
numero=$numfila}]
        set cb_ok3 [dpdg eval {SELECT ok3 FROM preguntas WHERE
numero=$numfila}]
        set cb_ok4 [dpdg eval {SELECT ok4 FROM preguntas WHERE
numero=$numfila}]
        set anulada [dpdg eval {SELECT anulada FROM preguntas WHERE
numero=$numfila}]
        set obsoleta [dpdg eval {SELECT obsoleta FROM preguntas WHERE
numero=$numfila}]
        # si vamos por esta rama le doy a markdown el valor que le corresponda
        if {$mkyes} {set markdown [dpdg eval {SELECT markdown FROM preguntas
WHERE numero=$numfila}]; .editar.ftitulo.cb_markdown configure -state normal} \
        else {.editar.ftitulo.cb_markdown configure -state disabled}
        #$preg edit modified 0
        $preg replace 1.0 end $enunciado
    }
    # set lectura 1
    #puts "pleer-lectura: $lectura"
}

```

```

#### procedimientos para activar y desactivar trazas
proc desactivar {} {
    global cb_ok1 cb_ok2 cb_ok3 cb_ok4
    global anulada obsoleta nombreTema enunciado mkyes markdown
    global respuesta1 respuesta2 respuesta3 respuesta4

    trace remove variable cb_ok1 write tracer
    trace remove variable cb_ok2 write tracer
    trace remove variable cb_ok3 write tracer

```

```

trace remove variable cb_ok4 write tracer
trace remove variable anulada write tracer
trace remove variable obsoleta write tracer
trace remove variable markdown write tracer
#trace remove variable enunciado write tracer
trace remove variable nombreTema write tracer
trace remove variable respuesta1 write tracer
trace remove variable respuesta2 write tracer
trace remove variable respuesta3 write tracer
trace remove variable respuesta4 write tracer
.editar.fnumero.bguardar configure -state disabled
# puts "desac"
}

proc activar {} {
    global cb_ok1 cb_ok2 cb_ok3 cb_ok4
    global anulada obsoleta nombreTema enunciado mkyes markdown
    global respuesta1 respuesta2 respuesta3 respuesta4

    trace add variable cb_ok1 write tracer
    trace add variable cb_ok2 write tracer
    trace add variable cb_ok3 write tracer
    trace add variable cb_ok4 write tracer
    trace add variable anulada write tracer
    trace add variable obsoleta write tracer
    trace add variable markdown write tracer
    #trace add variable enunciado write tracer
    trace add variable nombreTema write tracer
    trace add variable respuesta1 write tracer
    trace add variable respuesta2 write tracer
    trace add variable respuesta3 write tracer
    trace add variable respuesta4 write tracer
}

# aquí meto p.guardar, el procedimiento que se ejecuta al pulsar el botón de
guardar
# para guardar los nuevos valores de una pregunta
# habrá que chequear que lo que se guarda está bien, pero de momento supongo que
sí
proc p.guardar {} {
    global preg guardar numero
    global respuesta1 respuesta2 respuesta3 respuesta4
    global cb_ok1 cb_ok2 cb_ok3 cb_ok4 anulada obsoleta mkyes markdown
    global npflag nuevapreg
    global maxfilas
    global dicIdTemas
    #no se si hace falta lo siguiente
    global enunciado

    set enunciado [string trim [$preg get 1.0 end]]
    set idTema [dict get $dicIdTemas [.editar.ftitulo.ctitulo get]];# podría
chequear que la clave existe
    set todoOk 1;# luego aquí habrá que meter código para chequear que las
entradas están ok
    if {$todoOk} then {
        $guardar configure -state disabled
        # puts "npflag= $npflag"
        if {! $npflag} \
            then {
                dpdg eval {UPDATE preguntas SET \
                    tema=$idTema, \
                    pregunta=$enunciado, \
                    respuesta1=$respuesta1, \

```

```

        respuesta2=$respuesta2, \
        respuesta3=$respuesta3, \
        respuesta4=$respuesta4, \
        ok1=$cb_ok1, \
        ok2=$cb_ok2, \
        ok3=$cb_ok3, \
        ok4=$cb_ok4, \
        anulada=$anulada, \
        obsoleta=$obsoleta \
        WHERE numero=$numero }
    if {$mkyes} {dpdg eval {UPDATE preguntas SET markdown=$markdown
WHERE numero=$numero} }
} \
else {
    puts $enunciado
    if {$mkyes} then {
        dpdg eval {INSERT INTO preguntas \
        (tema,pregunta,\
        respuesta1,respuesta2,respuesta3,respuesta4,\
        ok1,ok2,ok3,ok4,\
        anulada,obsoleta,markdown)\
        VALUES \
        ($idTema,$enunciado,\
        $respuesta1,$respuesta2,$respuesta3,$respuesta4,\
        $cb_ok1,$cb_ok2,$cb_ok3,$cb_ok4,\
        $anulada,$obsoleta,$markdown)
        }
    } else {
        dpdg eval {INSERT INTO preguntas \
        (tema,pregunta,\
        respuesta1,respuesta2,respuesta3,respuesta4,\
        ok1,ok2,ok3,ok4,\
        anulada,obsoleta)\
        VALUES \
        ($idTema,$enunciado,\
        $respuesta1,$respuesta2,$respuesta3,$respuesta4,\
        $cb_ok1,$cb_ok2,$cb_ok3,$cb_ok4,\
        $anulada,$obsoleta)
        }
    }
    set npflag 0
    $nuevapreg configure -state active
    incr maxfilas
}
# a ado tentativamente lo siguiente pq parece que si no da error tras
guardar una pregunta
# lo quito tentativamente tambi n
# $guardar configure -state active
$preg edit modified 0
}
}

```

aqu  deber a poner un procedimiento p.comprobar, para chequear que lo que se guarda est  bien

aqu  meto p.nueva el procedimiento que se ejecuta cuando creo una nueva pregunta
esa nueva pregunta habr  que guardarla usando el comando guardar, si no se pierde

```

proc p.nueva {} {
    global maxfilas numero nuevapreg npflag

    $nuevapreg configure -state disabled

```

```
set npflag 1
set numero [expr $maxfilas + 1]
puts "numero = $numero"
p.leerfilas $numero
```

```
}
```

```

gTest <-
function
(fichDB, tipo=c("test", "final20", "final15"), temas=1, dias=0, copias=1, titulo) {
  ## fichDB es un string con el fichero .db (el sqlite con las preguntas)
  ## solo sirve para pasárselo al archivo .Rnw
  ## tipo es el tipo del examen
  ## temas es un vector con los números de los temas
  ## dias es el nº de días a partir de hoy para la fecha
  ## del examen
  ## copias es el número de test distintos que se generan
  ## titulo es simplemente para el encabezamiento del documento generado

  ## los parámetros que se pasan al fichero de latex son:
  ## - nombre_fichero_DB
  ## - encabezamiento_examen
  ## - dias
  ## - vtemas

  require(knitr)
  require(DBI)
  require(RSQLite)

  nombre_fichero_DB <- as.character(fichDB)
  ante_encabezamiento <- "Test de "
  ## encabezamiento_examen <- as.character(titulo)
  ## print(nombre_fichero_DB)

  ## esto genera un string para la consulta SQL
  ltemas <- "("
  for (tt in 1:length(temas))
    ltemas <- paste(ltemas, "temas.numero = ", temas[tt], " OR")
  ltemas <- substr(ltemas, 1, nchar(ltemas)-2)
  ltemas <- paste(ltemas, " ")
  vtemas <- ltemas
  ## print(vtemas)

  ## nombre del fichero .Rnw que uso como plantilla
  ## por defecto es el de test
  if (missing(tipo)) { plantilla <- "plantillaTest.Rnw"}
  else {
    tipo <- match.arg(tipo)
    if (tipo == "final20") {
      plantilla <- "plantillaFinal20.Rnw"
      ante_encabezamiento <- "Examen Final de "
    }
    else {
      if (tipo == "final15") {
        plantilla <- "plantillaFinal15.Rnw"
        ante_encabezamiento <- "Examen Final de "
      }
      else {plantilla <- "plantillaTest.Rnw"}
    }
  }
}

## esto extrae el nombre del fichero .Rnw, sin extensión
plantilla_sin_ext<- gsub(".\\w*$", "", plantilla, perl=T)

## esto extrae el nombre del fichero .db, sin extensión
## queda pendiente ver qué se hace en windows con los caracteres "\\"
## para la ruta de los archivos
asignatura_sin_ext <- tail(strsplit(nombre_fichero_DB, "/")[[1]], 1)
asignatura_sin_ext <- gsub(".\\w*$", "", asignatura_sin_ext, perl=T)
asignatura_sin_ext <- gsub("^test", "", asignatura_sin_ext, perl=T)

```



```

## print(asignatura_sin_ext)

if (missing(titulo)) {
  encabezamiento_examen <- paste0(ante_encabezamiento,"Gestión de
",asignatura_sin_ext)
}
else {
  encabezamiento_examen <- as.character(titulo)
}

for (kk in 1:copias) {
  knit(plantilla)

  ## print(class(fich))
  ## esto nos demuestra que fich es un string
  ## (el comentario anterior creo que ya no pinta nada)

  system (paste0("pdflatex ",plantilla_sin_ext,".tex"))
  ## print (paste0("pdflatex ",fichero,".tex"))

  system(paste0("rm ",plantilla_sin_ext,".tex"))
  system(paste0("rm ",plantilla_sin_ext,".aux"))
  system(paste0("rm ",plantilla_sin_ext,".log"))

  system(paste0("mv ",plantilla_sin_ext,".pdf ","ex",kk,".pdf"))
}
rm(kk)
}

```

```

\documentclass{article}
\usepackage[top=1in, bottom=1in, left=1in, right=1in]{geometry}
\usepackage[spanish]{babel}
\selectlanguage{spanish}
\usepackage[utf8]{inputenc}
\date{}
\begin{document}
\title{\Sexpr{encabezamiento_examen} (\Sexpr{format(Sys.Date()+dias,
"%d/%b/%Y")}))}
\maketitle

```

```

%\noindent
\textbf{Apellidos: \hspace{220pt} Nombre: }
\newline
\rule{\textwidth}{0.5pt}
\newline
%\newline

```

```

\textbf{DNI: \hspace{252pt} Grupo: }
\newline
\rule{\textwidth}{0.5pt}
\newline
\newline

```

```

<<echo=FALSE,message=FALSE,results="hide">>=
drv<-dbDriver("SQLite")
cat(nombre_fichero_DB)
con<-dbConnect(drv,dbname=nombre_fichero_DB)
mmm<-dbGetQuery(con,paste("SELECT pregunta, respuesta1, respuesta2, respuesta3,
respuesta4, anulada, obsoleta, markdown FROM preguntas JOIN temas ON tema=id
WHERE ", vtemas," AND ((NOT anulada) AND (NOT obsoleta) AND (NOT markdown))
ORDER BY RANDOM() LIMIT 10;"))
mmm<-mmm[,1:5]
for (rr in 1:10) {
  mmm[rr,2:5]<-mmm[rr,sample(2:5),4]
}
for (ii in 1:10) {
  for (jj in 1:5) {
    mmm[ii,jj]<-gsub("'",'\'',mmm[ii,jj],fixed=TRUE)
    mmm[ii,jj]<-gsub("%","\%",mmm[ii,jj],fixed=TRUE)
  }
}
rm(rr,ii,jj)
dbDisconnect(con)
dbUnloadDriver(drv)
@

```

```

\begin{enumerate}

\item \textbf{\Sexpr{mmm[1,1]}}
\begin{enumerate}
\item \Sexpr{mmm[1,2]}
\item \Sexpr{mmm[1,3]}
\item \Sexpr{mmm[1,4]}
\item \Sexpr{mmm[1,5]}
\end{enumerate}


```

```

\item \textbf{\Sexpr{mmm[2,1]}}
\begin{enumerate}

```

```

\item \Sexpr{mmm[2,2]}
\item \Sexpr{mmm[2,3]}
\item \Sexpr{mmm[2,4]}
\item \Sexpr{mmm[2,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[3,1]}}
\begin{enumerate}
\item \Sexpr{mmm[3,2]}
\item \Sexpr{mmm[3,3]}
\item \Sexpr{mmm[3,4]}
\item \Sexpr{mmm[3,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[4,1]}}
\begin{enumerate}
\item \Sexpr{mmm[4,2]}
\item \Sexpr{mmm[4,3]}
\item \Sexpr{mmm[4,4]}
\item \Sexpr{mmm[4,5]}
\end{enumerate}

```

\pagebreak

```

\item \textbf{\Sexpr{mmm[5,1]}}
\begin{enumerate}
\item \Sexpr{mmm[5,2]}
\item \Sexpr{mmm[5,3]}
\item \Sexpr{mmm[5,4]}
\item \Sexpr{mmm[5,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[6,1]}}
\begin{enumerate}
\item \Sexpr{mmm[6,2]}
\item \Sexpr{mmm[6,3]}
\item \Sexpr{mmm[6,4]}
\item \Sexpr{mmm[6,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[7,1]}}
\begin{enumerate}
\item \Sexpr{mmm[7,2]}
\item \Sexpr{mmm[7,3]}
\item \Sexpr{mmm[7,4]}
\item \Sexpr{mmm[7,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[8,1]}}
\begin{enumerate}
\item \Sexpr{mmm[8,2]}
\item \Sexpr{mmm[8,3]}
\item \Sexpr{mmm[8,4]}
\item \Sexpr{mmm[8,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[9,1]}}
\begin{enumerate}

```

```
\item \Sexpr{mmm[9,2]}
\item \Sexpr{mmm[9,3]}
\item \Sexpr{mmm[9,4]}
\item \Sexpr{mmm[9,5]}
\end{enumerate}
```

```
\item \textbf{\Sexpr{mmm[10,1]}}
\begin{enumerate}
\item \Sexpr{mmm[10,2]}
\item \Sexpr{mmm[10,3]}
\item \Sexpr{mmm[10,4]}
\item \Sexpr{mmm[10,5]}
\end{enumerate}
```

```
\end{enumerate}
```

```
\end{document}
```

```

\documentclass[11pt]{article}
\usepackage[top=0.7in, bottom=1in, left=1in, right=1in]{geometry}
\usepackage[spanish]{babel}
\selectlanguage{spanish}
\usepackage[utf8]{inputenc}
\usepackage{nicefrac}
\usepackage{tikz}
\date{}

\begin{document}
\title{\Sexpr{encabezamiento_examen} (\Sexpr{format(Sys.Date()+dias,
"%d/%b/%Y")} )}}
\maketitle

%\noindent
\textbf{Apellidos: \hspace{220pt} Nombre: }
\newline
\rule{\textwidth}{0.5pt}
\newline
%\newline

\textbf{DNI: \hspace{252pt} Grupo: }
\newline
\rule{\textwidth}{0.5pt}
\newline
\newline
\vspace{1cm}

\textbf{\hspace{0.3cm}Nota del test \hspace{1cm} + \hspace{1.1cm} Nota de los
ejercicios \hspace{0.8cm} = \hspace{1cm} Nota del Examen}
\newline

\begin{tikzpicture}
%\fill[gray!20!white] (0,0) rectangle (3,3); \fill[gray!20!white] (6,0)
rectangle (9,3); \fill[gray!20!white] (12,0) rectangle (15,3);
\draw (0,0) rectangle (3,3); \draw (6,0) rectangle (9,3); \draw (12,0) rectangle
(15,3);
\end{tikzpicture}

\vspace{2cm}
\textbf{Normas del Examen:}

\begin{itemize}
\item Tanto el test como los ejercicios puntúan sobre 5. La nota del examen será
la suma
de ambas calificaciones.
\item Cada pregunta de test tiene una única respuesta válida. Esa respuesta se
marcará
con claridad. Las respuestas ambiguas o poco claras se considerarán erróneas.
\item Las respuestas erróneas en el test penalizan  $\frac{1}{3}$  (con lo que
tres mal cancelarían una bien). Las
preguntas en blanco no penalizan.
\item Los ejercicios se contestarán en el cuadernillo rayado. Todos puntúan lo
mismo. El
orden en el que se respondan es indiferente.
\end{itemize}

<<echo=FALSE,message=FALSE,results="hide">>=

drv<-dbDriver("SQLite")
cat(nombre_fichero_DB)
con<-dbConnect(drv,dbname=nombre_fichero_DB)

```

```

mmm<-dbGetQuery(con,paste("SELECT pregunta, respuesta1, respuesta2, respuesta3,
respuesta4, anulada, obsoleta, markdown FROM preguntas JOIN temas ON tema=id
WHERE ", vtemas," AND ((NOT anulada) AND (NOT obsoleta) AND (NOT markdown))
ORDER BY RANDOM() LIMIT 15;"))
mmm<-mmm[,1:5]
for (rr in 1:15) {
  mmm[rr,2:5]<-mmm[rr,sample(2:5),4]
}
for (ii in 1:15) {
  for (jj in 1:5) {
    mmm[ii,jj]<-gsub("'",'\'',mmm[ii,jj],fixed=TRUE)
    mmm[ii,jj]<-gsub("%","\\%",mmm[ii,jj],fixed=TRUE)
  }
}
rm(rr,ii,jj)
dbDisconnect(con)
dbUnloadDriver(drv)
@

```

\pagebreak

\begin{enumerate}

\item \textbf{\Sexpr{mmm[1,1]}}

\begin{enumerate}

\item \Sexpr{mmm[1,2]}

\item \Sexpr{mmm[1,3]}

\item \Sexpr{mmm[1,4]}

\item \Sexpr{mmm[1,5]}

\end{enumerate}

\item \textbf{\Sexpr{mmm[2,1]}}

\begin{enumerate}

\item \Sexpr{mmm[2,2]}

\item \Sexpr{mmm[2,3]}

\item \Sexpr{mmm[2,4]}

\item \Sexpr{mmm[2,5]}

\end{enumerate}

\item \textbf{\Sexpr{mmm[3,1]}}

\begin{enumerate}

\item \Sexpr{mmm[3,2]}

\item \Sexpr{mmm[3,3]}

\item \Sexpr{mmm[3,4]}

\item \Sexpr{mmm[3,5]}

\end{enumerate}

\item \textbf{\Sexpr{mmm[4,1]}}

\begin{enumerate}

\item \Sexpr{mmm[4,2]}

\item \Sexpr{mmm[4,3]}

\item \Sexpr{mmm[4,4]}

\item \Sexpr{mmm[4,5]}

\end{enumerate}

\item \textbf{\Sexpr{mmm[5,1]}}

```

\begin{enumerate}
\item \Sexpr{mmm[5,2]}
\item \Sexpr{mmm[5,3]}
\item \Sexpr{mmm[5,4]}
\item \Sexpr{mmm[5,5]}
\end{enumerate}

```

```

\pagebreak

```

```

\item \textbf{\Sexpr{mmm[6,1]}}
\begin{enumerate}
\item \Sexpr{mmm[6,2]}
\item \Sexpr{mmm[6,3]}
\item \Sexpr{mmm[6,4]}
\item \Sexpr{mmm[6,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[7,1]}}
\begin{enumerate}
\item \Sexpr{mmm[7,2]}
\item \Sexpr{mmm[7,3]}
\item \Sexpr{mmm[7,4]}
\item \Sexpr{mmm[7,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[8,1]}}
\begin{enumerate}
\item \Sexpr{mmm[8,2]}
\item \Sexpr{mmm[8,3]}
\item \Sexpr{mmm[8,4]}
\item \Sexpr{mmm[8,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[9,1]}}
\begin{enumerate}
\item \Sexpr{mmm[9,2]}
\item \Sexpr{mmm[9,3]}
\item \Sexpr{mmm[9,4]}
\item \Sexpr{mmm[9,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[10,1]}}
\begin{enumerate}
\item \Sexpr{mmm[10,2]}
\item \Sexpr{mmm[10,3]}
\item \Sexpr{mmm[10,4]}
\item \Sexpr{mmm[10,5]}
\end{enumerate}

```

```

\pagebreak

```

```

\item \textbf{\Sexpr{mmm[11,1]}}
\begin{enumerate}
\item \Sexpr{mmm[11,2]}
\item \Sexpr{mmm[11,3]}
\item \Sexpr{mmm[11,4]}
\item \Sexpr{mmm[11,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[12,1]}}
\begin{enumerate}
\item \Sexpr{mmm[12,2]}
\item \Sexpr{mmm[12,3]}
\item \Sexpr{mmm[12,4]}
\item \Sexpr{mmm[12,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[13,1]}}
\begin{enumerate}
\item \Sexpr{mmm[13,2]}
\item \Sexpr{mmm[13,3]}
\item \Sexpr{mmm[13,4]}
\item \Sexpr{mmm[13,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[14,1]}}
\begin{enumerate}
\item \Sexpr{mmm[14,2]}
\item \Sexpr{mmm[14,3]}
\item \Sexpr{mmm[14,4]}
\item \Sexpr{mmm[14,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[15,1]}}
\begin{enumerate}
\item \Sexpr{mmm[15,2]}
\item \Sexpr{mmm[15,3]}
\item \Sexpr{mmm[15,4]}
\item \Sexpr{mmm[15,5]}
\end{enumerate}

```

```

\end{enumerate}

```

```

\end{document}

```



```

\documentclass[11pt]{article}
\usepackage[top=0.7in, bottom=1in, left=1in, right=1in]{geometry}
\usepackage[spanish]{babel}
\selectlanguage{spanish}
\usepackage[utf8]{inputenc}
\usepackage{nicefrac}
\usepackage{tikz}
\date{}

\begin{document}
\title{\Sexpr{encabezamiento_examen} (\Sexpr{format(Sys.Date()+dias,
"%d/%b/%Y")} )}}
\maketitle

%\noindent
\textbf{Apellidos: \hspace{220pt} Nombre: }
\newline
\rule{\textwidth}{0.5pt}
\newline
%\newline

\textbf{DNI: \hspace{252pt} Grupo: }
\newline
\rule{\textwidth}{0.5pt}
\newline
\newline
\vspace{1cm}

\textbf{\hspace{0.3cm}Nota del test \hspace{1cm} + \hspace{1.1cm} Nota de los
ejercicios \hspace{0.8cm} = \hspace{1cm} Nota del Examen}
\newline

\begin{tikzpicture}
%\fill[gray!20!white] (0,0) rectangle (3,3); \fill[gray!20!white] (6,0)
rectangle (9,3); \fill[gray!20!white] (12,0) rectangle (15,3);
\draw (0,0) rectangle (3,3); \draw (6,0) rectangle (9,3); \draw (12,0) rectangle
(15,3);
\end{tikzpicture}

\vspace{2cm}
\textbf{Normas del Examen:}

\begin{itemize}
\item Tanto el test como los ejercicios puntúan sobre 5. La nota del examen será
la suma
de ambas calificaciones.
\item Cada pregunta de test tiene una única respuesta válida. Esa respuesta se
marcará
con claridad. Las respuestas ambiguas o poco claras se considerarán erróneas.
\item Las respuestas erróneas en el test penalizan  $\frac{1}{3}$  (con lo que
tres mal cancelarían una bien). Las
preguntas en blanco no penalizan.
\item Los ejercicios se contestarán en el cuadernillo rayado. Todos puntúan lo
mismo. El
orden en el que se respondan es indiferente.
\end{itemize}

<<echo=FALSE,message=FALSE,results="hide">>=

drv<-dbDriver("SQLite")
cat(nombre_fichero_DB)
con<-dbConnect(drv,dbname=nombre_fichero_DB)

```

```

mmm<-dbGetQuery(con,paste("SELECT pregunta, respuesta1, respuesta2, respuesta3,
respuesta4, anulada, obsoleta, markdown FROM preguntas JOIN temas ON tema=id
WHERE ", vtemas," AND ((NOT anulada) AND (NOT obsoleta) AND (NOT markdown))
ORDER BY RANDOM() LIMIT 20;"))
mmm<-mmm[,1:5]
for (rr in 1:20) {
  mmm[rr,2:5]<-mmm[rr,sample(2:5),4]
}
for (ii in 1:20) {
  for (jj in 1:5) {
    mmm[ii,jj]<-gsub("'",'\'',mmm[ii,jj],fixed=TRUE)
    mmm[ii,jj]<-gsub("%","\\%",mmm[ii,jj],fixed=TRUE)
  }
}
rm(rr,ii,jj)
dbDisconnect(con)
dbUnloadDriver(drv)
@

```

\pagebreak

\begin{enumerate}

\item \textbf{\Sexpr{mmm[1,1]}}

\begin{enumerate}

\item \Sexpr{mmm[1,2]}

\item \Sexpr{mmm[1,3]}

\item \Sexpr{mmm[1,4]}

\item \Sexpr{mmm[1,5]}

\end{enumerate}

\item \textbf{\Sexpr{mmm[2,1]}}

\begin{enumerate}

\item \Sexpr{mmm[2,2]}

\item \Sexpr{mmm[2,3]}

\item \Sexpr{mmm[2,4]}

\item \Sexpr{mmm[2,5]}

\end{enumerate}

\item \textbf{\Sexpr{mmm[3,1]}}

\begin{enumerate}

\item \Sexpr{mmm[3,2]}

\item \Sexpr{mmm[3,3]}

\item \Sexpr{mmm[3,4]}

\item \Sexpr{mmm[3,5]}

\end{enumerate}

\item \textbf{\Sexpr{mmm[4,1]}}

\begin{enumerate}

\item \Sexpr{mmm[4,2]}

\item \Sexpr{mmm[4,3]}

\item \Sexpr{mmm[4,4]}

\item \Sexpr{mmm[4,5]}

\end{enumerate}

\item \textbf{\Sexpr{mmm[5,1]}}

```

\begin{enumerate}
\item \Sexpr{mmm[5,2]}
\item \Sexpr{mmm[5,3]}
\item \Sexpr{mmm[5,4]}
\item \Sexpr{mmm[5,5]}
\end{enumerate}

```

```

\pagebreak

```

```

\item \textbf{\Sexpr{mmm[6,1]}}
\begin{enumerate}
\item \Sexpr{mmm[6,2]}
\item \Sexpr{mmm[6,3]}
\item \Sexpr{mmm[6,4]}
\item \Sexpr{mmm[6,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[7,1]}}
\begin{enumerate}
\item \Sexpr{mmm[7,2]}
\item \Sexpr{mmm[7,3]}
\item \Sexpr{mmm[7,4]}
\item \Sexpr{mmm[7,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[8,1]}}
\begin{enumerate}
\item \Sexpr{mmm[8,2]}
\item \Sexpr{mmm[8,3]}
\item \Sexpr{mmm[8,4]}
\item \Sexpr{mmm[8,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[9,1]}}
\begin{enumerate}
\item \Sexpr{mmm[9,2]}
\item \Sexpr{mmm[9,3]}
\item \Sexpr{mmm[9,4]}
\item \Sexpr{mmm[9,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[10,1]}}
\begin{enumerate}
\item \Sexpr{mmm[10,2]}
\item \Sexpr{mmm[10,3]}
\item \Sexpr{mmm[10,4]}
\item \Sexpr{mmm[10,5]}
\end{enumerate}

```

```

\pagebreak

```

```

\item \textbf{\Sexpr{mmm[11,1]}}
\begin{enumerate}
\item \Sexpr{mmm[11,2]}
\item \Sexpr{mmm[11,3]}
\item \Sexpr{mmm[11,4]}
\item \Sexpr{mmm[11,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[12,1]}}
\begin{enumerate}
\item \Sexpr{mmm[12,2]}
\item \Sexpr{mmm[12,3]}
\item \Sexpr{mmm[12,4]}
\item \Sexpr{mmm[12,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[13,1]}}
\begin{enumerate}
\item \Sexpr{mmm[13,2]}
\item \Sexpr{mmm[13,3]}
\item \Sexpr{mmm[13,4]}
\item \Sexpr{mmm[13,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[14,1]}}
\begin{enumerate}
\item \Sexpr{mmm[14,2]}
\item \Sexpr{mmm[14,3]}
\item \Sexpr{mmm[14,4]}
\item \Sexpr{mmm[14,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[15,1]}}
\begin{enumerate}
\item \Sexpr{mmm[15,2]}
\item \Sexpr{mmm[15,3]}
\item \Sexpr{mmm[15,4]}
\item \Sexpr{mmm[15,5]}
\end{enumerate}

```

\pagebreak

```

\item \textbf{\Sexpr{mmm[16,1]}}
\begin{enumerate}
\item \Sexpr{mmm[16,2]}
\item \Sexpr{mmm[16,3]}
\item \Sexpr{mmm[16,4]}
\item \Sexpr{mmm[16,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[17,1]}}
\begin{enumerate}
\item \Sexpr{mmm[17,2]}
\item \Sexpr{mmm[17,3]}
\item \Sexpr{mmm[17,4]}
\item \Sexpr{mmm[17,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[18,1]}}
\begin{enumerate}
\item \Sexpr{mmm[18,2]}
\item \Sexpr{mmm[18,3]}
\item \Sexpr{mmm[18,4]}
\item \Sexpr{mmm[18,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[19,1]}}
\begin{enumerate}
\item \Sexpr{mmm[19,2]}
\item \Sexpr{mmm[19,3]}
\item \Sexpr{mmm[19,4]}
\item \Sexpr{mmm[19,5]}
\end{enumerate}

```

```

\item \textbf{\Sexpr{mmm[20,1]}}
\begin{enumerate}
\item \Sexpr{mmm[20,2]}
\item \Sexpr{mmm[20,3]}
\item \Sexpr{mmm[20,4]}
\item \Sexpr{mmm[20,5]}
\end{enumerate}

```

```

\pagebreak

```

```

\end{enumerate}

```

```

\end{document}

```

7. Anexo 2: Tutorial de la ñapa

Se anexa a continuación un breve tutorial de la ñapa

Tutorial sobre la ñapa

Equipo Proyecto Innova Docencia (UCM) nº 335

28 de junio de 2020

1. Panorámica general

La ñapa es un pequeño programa escrito en el lenguaje TclTk que busca facilitar la gestión de bases de datos de preguntas de test en el *Learning Management System* Moodle. La interfaz web que ofrece Moodle para la introducción y gestión de preguntas de test resulta un tanto lenta, farragosa y proclive a dar lugar a problemas, y por esa razón, y para intentar evitar esos problemas, se ha diseñado la ñapa. El objetivo inicial era integrar la ñapa dentro de un paquete de R, ya que R permite con facilidad utilizar programas escritos en TclTk, pero de momento la ñapa es simplemente un programa independiente, aún no está integrado en un paquete de R. Pese a ello es un programa perfectamente útil y funcional.

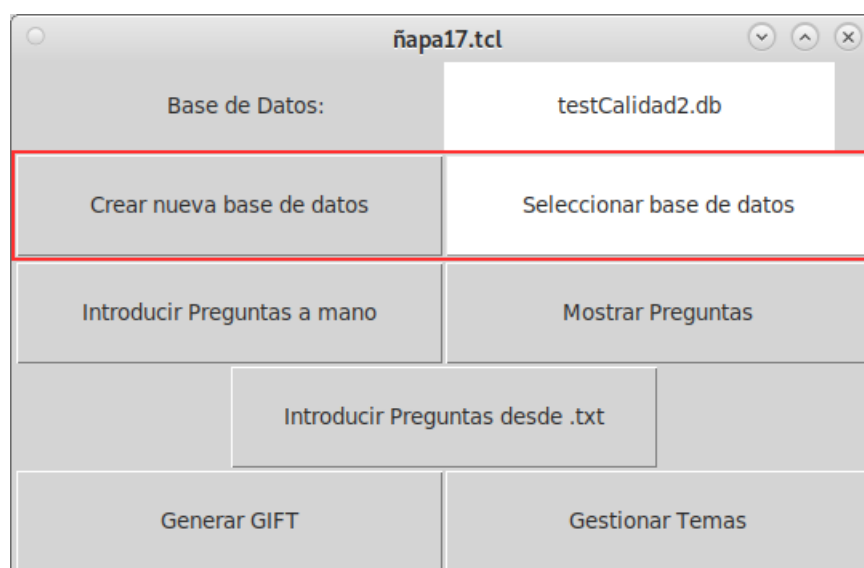
La idea básica para poder usar la ñapa es que se trata de un programa para gestionar bases de datos de preguntas de test contenidas en un archivo *sqlite*. *sqlite* es un gestor de bases de datos de software libre que ha tenido una enorme aceptación y se ha convertido en la base de datos más usada a nivel mundial. Se pueden tener varias bases de datos de preguntas de test de este tipo, por ejemplo para diferentes asignaturas, y desde la ñapa se puede seleccionar en cada momento con cuál se trabaja. En consecuencia para poder usar la ñapa es necesario tener como mínimo dos archivos, la propia ñapa y una base de datos *sqlite* con las preguntas de test. La ñapa permite editar y modificar estas preguntas con facilidad, visualizarlas y generar un archivo en formato *gift* con preguntas de test para un examen que puede ser directamente importado en Moodle.

El lenguaje en el que está escrita la ñapa, TclTk, es un lenguaje de *scripting*, lo que significa que no está compilado, sino que se va interpretando y ejecutando instrucción a instrucción, pero para poder ejecutarlo es necesario tener instalado el intérprete correspondiente. En linux y Mac el intérprete está fácilmente disponible, en windows se puede instalar, por ejemplo, el proporcionado por ActiveState (<https://www.activestate.com>).

com/products/tcl/downloads/). Una alternativa en windows consiste simplemente en usar una versión ejecutable de la ñapa (el típico programa de windows con extensión .exe). En el enlace <https://pepe.sds-ip.de/~pepe/> están disponibles tanto el script con el código fuente de la ñapa (con extensión .tcl y ejecutable directamente en cualquier sistema operativo con el intérprete de Tcl instalado) y el ejecutable para windows (con extensión .exe y que puede ser ejecutado sin necesidad de instalar nada). Este ejecutable para windows se ha generado usando freewrap (<http://freewrap.sourceforge.net/>).

2. Ventana principal

Cuando se ejecuta la ñapa aparece su ventana principal con una serie de opciones:



Arriba del todo se indica la base de datos que está seleccionada y sobre la que se va a trabajar. Será siempre una base de datos sqlite con extensión db y la estructura adecuada (formada por dos tablas, una con los temas y otra con las preguntas; en cada pregunta se indica a qué tema pertenece; cada tema tendrá un número y un título, que se podrán cambiar). Se puede cambiar la base de datos seleccionada usando el botón «Seleccionar base de datos» y se puede crear una nueva base de datos vacía con el botón «Crear nueva base de datos».

3. Gestión de temas

Si se pulsa el botón «Gestionar Temas» aparece una ventana como la siguiente, en la que se indican los temas que existen actualmente en la base de datos, con su número y

su nombre, cuantas preguntas válidas hay en cada tema y cuántas están marcadas como anuladas o como obsoletas.

temas					
id	Nº Tema	Título Tema	Preguntas Válidas	Preguntas Anuladas	Preguntas Obsoletas
8	1	Introducción	21	0	1
5	2	Gestión de la Calidad Total	29	2	0
11	3	Sistemas de Gestión de la Calidad	20	0	0
2	4	Control Estadístico de Procesos	20	0	0
13	5	Documentación	21	0	0
7	6	Herramientas de Gestión de la Calidad	20	2	0
9	7	Muestreo de Aceptación	20	0	5
1	8	Calidad Medioambiental	21	0	6
3	9	Diseño de Experimentos	15	0	0
4	10	Gestión de Calidad en Servicios	19	1	0
12	11	Técnicas de Gestión de la Calidad	8	0	0
6	12	Gurús	2	0	0
10	13	Premios de Excelencia	4	0	0
14	14	Calidad Turística España	20	0	0
15	15	Calidad Alimentaria	17	0	0
16	41	SPC_Ejercicios_1	6	0	0
17	42	SPC_Ejercicios_2	6	0	0
18	43	SPC_Ejercicios_3	6	0	0
19	44	SPC_Ejercicios_4	6	0	0
20	71	Muestreo_Ejercicios_1	6	0	0
21	72	Muestreo_Ejercicios_2	6	0	0
22	73	Muestreo_Ejercicios_3	6	0	0
23	74	Muestreo_Ejercicios_4	6	0	0
24	99	Test Turismo	34	0	0

Nuevo Tema

Guardar Cambios Cerrar

Ofrece también la posibilidad de introducir un nuevo tema, con su número y su título, tras lo cual habrá que darle al botón «Guardar Cambios». No existe todavía la posibilidad de eliminar un tema, pero sí se puede dejarlo con 0 preguntas moviéndolas a otro tema.

4. Introducción de preguntas

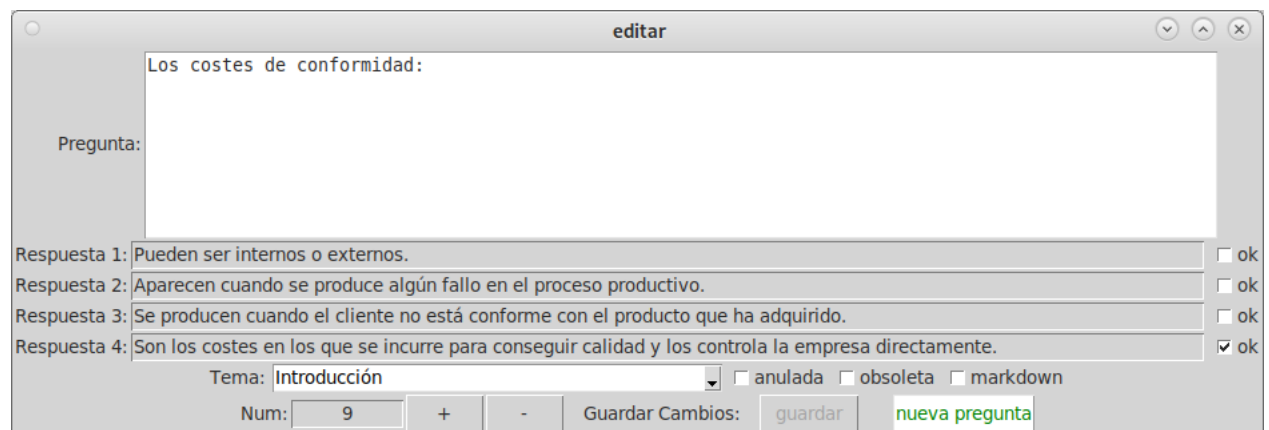
Las preguntas se pueden introducir en bloque o una a una.

Se introducen en bloque desde un archivo de texto plano (típicamente con extensión .txt). Este archivo se puede crear en word, por ejemplo, y luego guardarlo como .txt.

La estructura de este archivo es sencilla. Esta dividido en bloques, cada uno delimitado por una línea que contiene un texto como Tema 1, Tema 2, etc (la capitalización es indiferente) que indica el tema al que pertenecen las preguntas que van a continuación. Si el tema no existe en la base de datos se crea automáticamente.

Cada bloque está formado por una secuencia de enunciados de preguntas seguidos por sus cuatro posibles respuestas. Es decir, cada pregunta completa está formada por cinco líneas (entendiendo por línea una secuencia de caracteres hasta un Retorno de Carro, es decir, hasta el carácter que se introduce al pulsar Intro en el teclado). Estas líneas pueden ir separadas entre sí por líneas en blanco. Entonces la ñapa lee y procesa las líneas de 5 en 5 asumiendo que la primera del bloque de 5 es un enunciado de pregunta y las 4 siguientes sus cuatro respuestas. La o las respuestas correctas se indican poniendo un asterisco al principio o al final de la respuesta.

Por otra parte las preguntas se pueden introducir, o modificar, de una en una pulsando en el botón «Introducir preguntas a mano», que muestra una ventana como la siguiente:



The screenshot shows a window titled "editar" with a standard macOS-style title bar. Inside, there's a large text area for the question, labeled "Pregunta:" on the left. Below this, there are four rows for answers, each labeled "Respuesta 1:" through "Respuesta 4:". Each row has a text input field and a checkbox labeled "ok" on the right. The first three checkboxes are unchecked, and the fourth is checked. Below the answer fields, there's a "Tema:" dropdown menu currently set to "Introducción". To the right of the dropdown are three checkboxes: "anulada", "obsoleta", and "markdown", all of which are unchecked. At the bottom, there's a "Num:" field with the value "9", followed by "+" and "-" buttons. To the right of these is a "Guardar Cambios:" label, followed by two buttons: "guardar" and "nueva pregunta".

En esta ventana se puede seleccionar la pregunta que se quiere ver o editar por su número, se puede crear una nueva pregunta, y se puede introducir o editar el enunciado de la pregunta, las cuatro respuestas, seleccionar el tema al que pertenece la pregunta, indicar las respuestas correctas, o marcar la pregunta como anulada, obsoleta, o que está en formato *markdown*, que puede ser útil especialmente si se desea que en el enunciado haya una tabla.

Las preguntas existentes en cada tema se pueden ver pulsando el botón «Mostrar Preguntas», que no lleva a una ventana en la que se puede seleccionar el tema que se desea ver y muestra las preguntas del tema en un cuadro de texto (si se edita no tiene efecto sobre las preguntas).

5. Generación de archivos de preguntas en formato *gift* para Moodle

Pulsando el botón «Generar Gift» aparece una ventana con una serie de opciones para generar un archivo con preguntas en formato gift que es directamente importable en el banco de preguntas de Moodle:

The screenshot shows a window titled 'generar' with standard window controls. The main heading is 'GENERAR ARCHIVO .GIFT PARA MOODLE'. Below this, it says 'indique los temas que se incluirán:' followed by a grid of checkboxes for topics: 'tema 1' through 'tema 15', 'tema 41' through 'tema 43', 'tema 44', 'tema 71', 'tema 72', 'tema 73', 'tema 74', and 'tema 99'. Below the checkboxes are two text input fields: 'Preguntas a quitar:' with a red background and 'Preguntas a añadir:' with a green background. Under these fields, there are two radio button options for '% que restan las erróneas:': '25.0%' (selected) and '33.3%'. Below that is a checkbox for '¿Incluir quinta pregunta?:' with the text 'Actualmente No. Marcar para cambiar' next to it. At the bottom center is a button labeled 'Generar Fichero'.

Allí se puede seleccionar el o los temas cuyas preguntas se incluirán, y se pueden añadir o quitar preguntas adicionales indicada por su número, especificándolas en los cuadros de texto en verde (para añadir preguntas) o en rojo (para quitarlas); para ello simplemente se ponen los números de las preguntas separados por espacios en blanco.

También ofrece la posibilidad de seleccionar cuánto restan las respuestas erróneas (un 25 % o un 33 %) y ofrece además la posibilidad de incluir una quinta respuesta que sea «No se selecciona ninguna respuesta», útil cuando el test tiene una sola respuesta válida.

Pulsando al botón generar fichero se genera un archivo de texto plano que se puede importar directamente en Moodle.

6. Generación de exámenes en papel

SE pueden generar exámenes de test en formato pdf, para su uso presencial, no *on-line* a través del campus virtual. Sin embargo esta opción no está implementada en la ñapa,

sino a través de una función de R. Para ello hay que tener instalado R y copiar en un directorio los cuatro archivos `generarTest.R` (que contiene la función de R `gTest()`) y `plantillaTest.Rnw`, `plantillaFinal15.Rnw` y `plantillaFinal20.Rnw` (que son plantillas con tres formatos de exámenes, de 10, de 15 y de 20 preguntas). Estos archivos están también disponibles en el enlace <https://pepe.sds-ip.de/~pepe/>.

Una vez hecho esto se ejecuta R, se carga el archivo `generarTest.R` con el comando:

```
> source("generarTest.R")
```

y posteriormente ya se pueden generar los test invocando a la función `gTest()`, que admite una serie de argumentos:

- `fichDB`, que es obligatorio, es el nombre (con su extensión) del fichero sqlite con las preguntas (va entre comillas)
- `tipo` es el tipo del examen, que será "test", el de 10 preguntas y es el valor por defecto, "final15" y "final20"
- `temas` es un vector de R con los temas cuyas preguntas se incluirán; p, ej si se quiere hacer un test con preguntas de los temas 4 y 6 habrá que poner `temas=c(4,6)`
- `días` sirve para indicar la fecha del examen, contado en días a partir de hoy (por defecto es cero)
- `copias` indica el número de variantes del test (por defecto es 1)
- `titulo` (sin acento), es un texto entrecomillado opcional pero recomendable con el nombre que se le pondrá al examen.